Technical Note 1971-10

C. H. Much
N. P. Smith
F. W. Floyd
E. H. Swenson

LES-8/9 Attitude Control System Numerical Simulation

5 February 1971

Prepared under Electronic Systems Division Contract F19628-70-C-0230 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Lexington, Massachusetts

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

LES-8/9 ATTITUDE CONTROL SYSTEM NUMERICAL SIMULATION

C. H. MUCH
N. P. SMITH

Group 76

F. W. FLOYD
E. H. SWENSON

Group 63

TECHNICAL NOTE 1971-10

5 FEBRUARY 1971

LEXINGTON MASSACHUSETTS

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, with the support of the Department of the Air Force under Contract F19628-70-C-0230.

ABSTRACT

A program was written to simulate the three axis attitude control system of LES-8/9. The underlying theory to the computer simulation and detailed outlines of each of the subroutines involved in the complete program are described.

Whenever feasible, the simulation was written to duplicate as closely as possible the logic and signal flow of the actual digital attitude control system. Each subroutine was thoroughly verified as accurate by independent and integral system operation, as well as by theoretical estimates, analog computer simulations and actual experimental data.

A substantial compilation of data from this working program was catalogued and analyzed for attitude control system evaluation and optimization. This program also proved itself to be invaluable in the analysis of stability and performance of the complete attitude control system.

Accepted for the Air Force
Joseph R. Waterman, Lt. Col., USAF
Chief, Lincoln Laboratory Project Office

CONTENTS

NOMENCLATURE

Z_1, Z_2, Z_3	principal axes of satellite body designated pitch, roll, yaw axes, respectively
Y_1	Euler angle representing pitch axis pointing error
Y_2	Euler angle representing roll axis pointing error
Y_3	Euler angle representing yaw axis pointing error
Y_4	momentum wheel gimbal angle
Y_5	angular momentum stored in reaction wheel (wheel speed)
Y_6	rate about pitch axis
Y_7	rate about roll axis
Y_8	rate about yaw axis
K_{qr}	quantization scale factor for roll axis control law (ft-lbs/bit)
S_1	sensor measured angle representing pitch axis pointing error
S_2	sensor measured angle representing roll axis pointing error
XJ_1, XJ_2, XJ_3	satellite principal inertias about pitch, roll, yaw axes, respectively
XJ_4	inertia of reaction wheel rotor about spin axis
M2, M3	misalignment (offset) angles between gimbal axis and satellite principal axis
IRNR	roll axis control law input to gimbal power amplifier
IRNP	pitch axis control law input to momentum wheel speed control logic
L1, L2, L3	pulsed plasma thruster control torques applied along the axes Z_1, Z_2, Z_3, respectively
LSB	momentum wheel speed period quantization in parts per second
XD1	gimbal damping coefficient
XD2	gimbal flex pivot spring constant
TD1, TD2, TD3	external disturbance torques applied along the Z_1, Z_2, Z_3 axes, respectively
SIGMA	standard deviation (rms value) of random noise generated by earth sensor
XP1, XP2, NP	pitch axis control law parameters
XR1, XR2, NR	roll axis control law parameters
T1, T2	reaction control torque of momentum wheel about its spin axis in the on-off state, respectively
N	update period or interval between sampling times

LES-8/9 ATTITUDE CONTROL SYSTEM SIMULATION

I. INTRODUCTION

The numerical simulation of the motion of a rigid body satellite containing a single gimbaled momentum wheel about its center of gravity is described. Also included is a simulation of the digital logic flow of the various attitude control laws, IR earth sensors and pulsed plasma thrusters.

A brief description of the satellite equations of motion and kinematics is included in Sec. II. An Adams-Moulton, Runge-Kutta integration subroutine[1] was used to solve the basic body dynamic equations. This simulation is intended for the analysis of relatively short real time attitude motion of the satellite on the order of a few minutes. Required integration time is large enough in this program so that the long term steady state response to very low frequency disturbances such as solar pressure torques are more economically determined in a separate simulation not included here. The exact kinematics are included so that both small and large angle attitude motion can be simulated with this program.

The various parts of the simulation are broken into 13 subroutines, each of which is briefly described in Sec. IV of this report. Required input data and typical output data for the program are given in Sec. V. A listing of the program is given in the Appendix. The attitude control laws for each operating mode of the pitch and roll axis systems are described in Sec. III.

II. SYSTEM EQUATIONS

The Euler equations of motion for a rigid body satellite containing a single gimbaled momentum wheel are given in Ref. 2, along with the assumptions used in the derivation. These equations are solved in the numerical simulation to provide the satellite body rates. The body rates are then converted to Euler angle rates referenced to an orbital reference frame. The Euler angle rates are then integrated to provide an inertial attitude reference for the satellite. The Euler angles are finally converted to earth sensor measured angles which are the true inputs to the attitude control system.

A. Euler Equations of Motion

$$\dot{Y}_4 = -\frac{1}{XD_1}\{(XD_2)\,Y_4 + Y_5\,[Y_8 + (Y_4 + M_2)\,(Y_6 - M_3Y_7)]\} - K_{qr}(IRNR) \tag{1}$$

$$\dot{Y}_5 = T_M \tag{2}$$

$$\dot{Y}_6 = \frac{1}{XJ_1}\,[L_1 + TD_1 + (XJ_{23})\,Y_7Y_8 + (Y_4 + M_2)\,Y_5(\dot{Y}_4 + Y_7) - T_M - M_3Y_5Y_8] \tag{3}$$

$$\dot{Y}_7 = \frac{1}{XJ_2}\,[L_2 + TD_2 + (XJ_{31})\,Y_6Y_8 - (Y_4 + M_2)\,Y_5(Y_6 + M_3\dot{Y}_4) + M_3T_M - Y_5Y_8] \tag{4}$$

$$\dot{Y}_8 = \frac{1}{XJ_2}\,[L_3 + TD_3 - (XJ_{21})\,Y_6Y_7 + Y_5(\dot{Y}_4 + Y_7) + (Y_4 + M_2)\,T_M + M_3Y_5Y_6] \quad . \tag{5}$$

Conversion from a satellite reference frame to an orbit reference frame is accomplished using the standard Euler angle transformation with Y_1, Y_2, Y_3 taken about the axes Z_1, Z_2, Z_3 in this order, respectively.

B. Euler Angle Transformations[3]

$$\dot{Y}_1 = [Y_6 \cos Y_3 - Y_7 \sin Y_3]/\cos Y_2 - \omega_o \tag{6}$$

$$\dot{Y}_2 = Y_6 \sin Y_3 + Y_7 \cos Y_3 \tag{7}$$

$$\dot{Y}_3 = Y_8 - \tan Y_2 [Y_6 \cos Y_3 - Y_7 \sin Y_3] \quad . \tag{8}$$

C. Earth Sensor Measured Angles

$$S_1 = \tan^{-1} [\sin Y_1 \cos Y_3 + \cos Y_1 \sin Y_3 \sin Y_2]/\cos Y_1 \cos Y_2 \tag{9}$$

$$S_2 = \tan^{-1} [\cos Y_1 \sin Y_2 \cos Y_3 - \sin Y_1 \sin Y_3]/\cos Y_1 \cos Y_2 \quad . \tag{10}$$

D. Earth Sensor Simulation

Relating the earth pitch axis sensor measured angle S_1 to the output (ISAMP) of the pitch sensor at sampling intervals of (IS) seconds yields

$$\text{ISAMP(IS)} = [S_1/\text{DELTAR} + \text{RMS}] \quad , \tag{11}$$

where

DELTAR = least significant bit of sensor quantized output (rad/bit)

RMS = quantized Gaussian distributed random noise generated by sensor with zero mean and standard deviation SIGMA

IS = sampling interval = 1, 2, . . . 16.

At periods of 16 sampling intervals the value of IS is reset to zero and the stored values of ISAMP(IS) are averaged to yield the sensor output ISP.

$$\text{ISP} = -\frac{1}{16} \sum_{I=1}^{16} \text{ISAMP(I)} \quad . \tag{12}$$

The value of ISP is updated once for every 16 samples of ISAMP. Since ISAMP is sampled once every DELTAT seconds, the value of ISP is updated once every N seconds, where N = (16) DELTAT.

The roll sensor is simulated identically to the pitch sensor. Repeating Eqs. (11) and (12) with roll sensor variables yields

$$\text{ISAMR(IS)} = [S_2/\text{DELTAR} + \text{RMSR}] \tag{13}$$

$$\text{ISR} = \frac{1}{16} \sum_{I=1}^{16} \text{ISAMR(I)} \quad . \tag{14}$$

Thus, the quantized output of the pitch and roll earth sensors are named ISP and ISR, respectively.

The linear field of view limitation on both sensors is included in the simulation. The total field of view is not terminated abruptly at the end of the linear range, but extends in a piecewise linear symmetrical configuration called a "foldover" characteristic. This characteristic effectively doubles the acquisition field of view of the control system. Also included here is the effect of roll attitude error on the pitch sensor field of view and vice versa.

III. ATTITUDE CONTROL LAWS

The attitude control laws for the pitch and roll axis systems are broken into two categories: (a) momentum exchange and (b) momentum expulsion. In this program there are several control modes simulated for both pitch and roll axis control. A separate control law is described for each operating mode in the attitude control system.

A. Pitch Axis Control Modes

There are 5 operating modes available for controlling the satellite attitude about the pitch axis. Four modes utilize momentum exchange capability of the reaction wheel and one uses the pulsed plasma thruster for direct momentum expulsion attitude control. The control laws utilized in these 5 operating modes are described.

1. Momentum Wheel Speed Control

The control laws relating to the motor torque T_M and momentum Y_5 to the wheel speed reference input IRNP are given in this section. The actual relative wheel speed period T_P is simulated from the equation

$$T_P = 2\pi/[Y_5/XJ_4 - Y_6 + Y_4Y_8] \quad . \tag{15}$$

The control system determines the wheel speed period T_P by measuring the interval between 8 successive tachometer pulses. The value of T_P is quantized into LSB bits per second and is updated at the rate of once per period. The digital number used to represent T_P is given by IPNP, where

$$\text{IPNP} = T_P(\text{LSB})\ \text{bits} \quad . \tag{16}$$

The value of IPNP is compared once per period with the wheel speed command IRNP to determine the error in the loop.

$$\text{IERROR} = (\text{IPNP} - \text{IRNP}) \quad . \tag{17}$$

The momentum wheel speed is binary controlled by either applying a constant reference voltage or zero to the motor windings according to the control law:

$$\begin{aligned} &\text{If} \quad (\text{ERROR} \leqslant 0) \quad , \quad T_M = -T_2 \\ &\text{If} \quad (\text{ERROR} > 0) \quad , \quad T_M = T_1 \quad . \end{aligned} \tag{18}$$

This binary control loop is updated once per period.

The reaction torque values T_1, T_2 used in the simulation closely approximate the actual nonlinear wheel control torques. This is accomplished by least-squares, fitting a first degree function to a set of experimentally determined torque values from the actual wheel.

$$\begin{aligned} &T_1 = A_1 + B_1\omega \quad , \quad \omega = \text{momentum wheel speed} = \frac{Y_5}{XJ_4} \\ &T_2 = A_2 + B_2\omega \quad . \end{aligned} \tag{19}$$

The coefficients A_1, A_2 are determined in a one g environment and to relate them to a zero g field requires a reduction in their value to compensate for the reduction in coulomb friction in space. The coefficients B_1, B_2 will vary slightly with wheel temperature. This is taken into account when simulating various conditions in which the wheel operates.

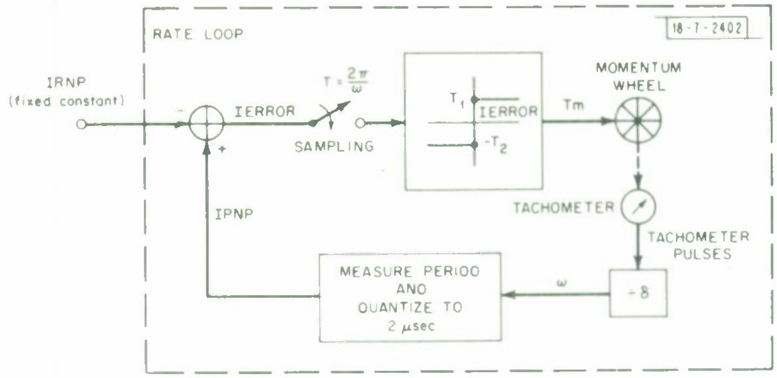

Fig. 1. Mode P1: Constant wheel speed.

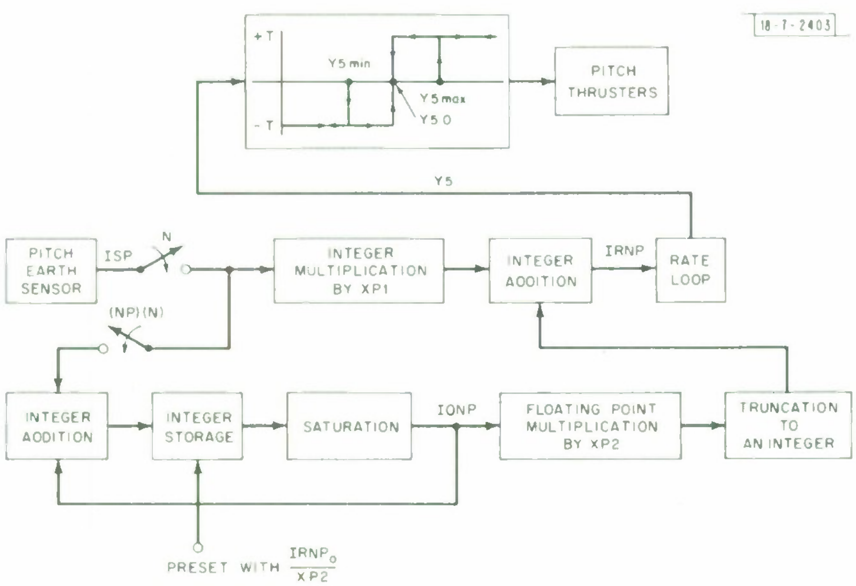

Fig. 2. Mode P2: Normal pitch control.

Since the momentum wheel always operates at a positive bias speed ω_o, the value of IRNP will be limited within a fixed range of values for all operating modes.

2. Mode P1

In mode P1 (constant wheel speed) shown in Fig. 1, the input to the wheel speed system IRNP is fixed at a constant value. In this mode the wheel speed is held at its initial condition value unless commanded to a different speed.

3. Mode P2

In mode P2 (normal pitch control) shown in Fig. 2, the input IRNP to the wheel speed system is a function of the earth sensor output ISP. The control law for this mode uses momentum exchange to control the pitch pointing. In addition, a coarse momentum expulsion control law is used to regulate wheel speed within fixed upper and lower bounds to prevent saturation. The mode P2 control equations are given below.

$$\mathrm{IRNP} = \mathrm{ISP(XP1)} + \mathrm{IQNP(XP2)} \quad . \tag{20}$$

This relation is updated at intervals of N seconds when the value of ISP is updated. The constants XP1, XP2 are parameters whose values are adjusted to yield optimum pitch performance in mode P2.

$$\mathrm{IQNP} \triangleq \sum_{(N)NP} \mathrm{ISP} \quad , \quad NP = \text{integer} \quad , \quad N = 4\,\text{sec} \quad . \tag{21}$$

Relation (21) is updated at intervals of (N)(NP) seconds. The parameter NP is also adjusted to optimize pointing performance in conjunction with XP1, XP2. The value of IQNP is limited to the bounds

$$\mathrm{QNPMIN} \leqslant \mathrm{IQNP} \leqslant \mathrm{QNPMAX} \tag{22}$$

to keep the wheel speed from saturating.

4. Mode P3

Mode P3 is termed the "sun acquisition" mode. In this mode shown in Fig. 3, the output from a wide angle sun sensor is used to automatically stop large spin rates about the pitch axis

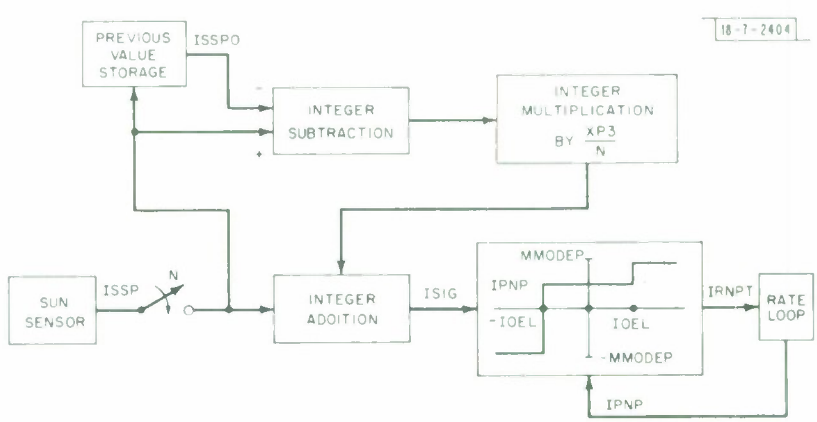

Fig. 3. Mode P3: Sun acquisition mode.

and point a reference axis to the sun within one degree. This mode uses the sun sensor reference S_1 as an input to the mode P3 control law logic. This mode is a momentum exchange law which uses the reaction wheel for its torque source. The quantized output, ISSP, of the sun reference sensor is given by the relation

$$\mathrm{ISSP} = [-S_1/\mathrm{DELTS}] \quad , \tag{23}$$

where DELTS = least significant bit of quantized sun sensor output.

The mode P3 control law generates a proportional plus derivative switching function, ISIG. When ISIG is less than a threshold value, IDEL, the momentum wheel is run at constant speed. When ISIG exceeds IDEL, the wheel speed is increased or decreased based on the sign of ISIG,

where

$$\mathrm{ISIG} = \frac{\mathrm{XP3}}{N}\,[\mathrm{ISSP} - \mathrm{ISSPO}] + \mathrm{ISSP}$$

XP3 = constant design parameter controlling the amount of damping in the system

$$\mathrm{ISSPO} = \text{value of ISSP at the previous sampling interval} \quad . \tag{24}$$

The function ISIG is updated at intervals of N seconds and is nulled about a threshold value IDEL with the control law:

$$\text{If} \quad |\mathrm{ISIG}| > \mathrm{IDEL} \quad , \quad \mathrm{IRNPT} = (\mathrm{MMODEP})\ \mathrm{sgn}(\mathrm{ISIG}) \tag{25}$$

$$\text{If} \quad |\mathrm{ISIG}| \leqslant \mathrm{IDEL} \quad , \quad \mathrm{IRNPT} = \mathrm{IPNP} \quad , \tag{26}$$

where

IDEL = quantized threshold value of mode P3 pointing accuracy

MMODEP = large constant representing an overriding input to the wheel speed control loop which effectively uncouples the wheel speed feedback

IRNPT = quantized input to the wheel speed control system, replacing the quantity IRNP used in mode P2.

5. Mode P4

Mode P4 is termed the pitch axis "back-up" mode. In this mode shown in Fig. 4, the ternary control law is similar in nature to that of mode P3, but now the system is a momentum expulsion

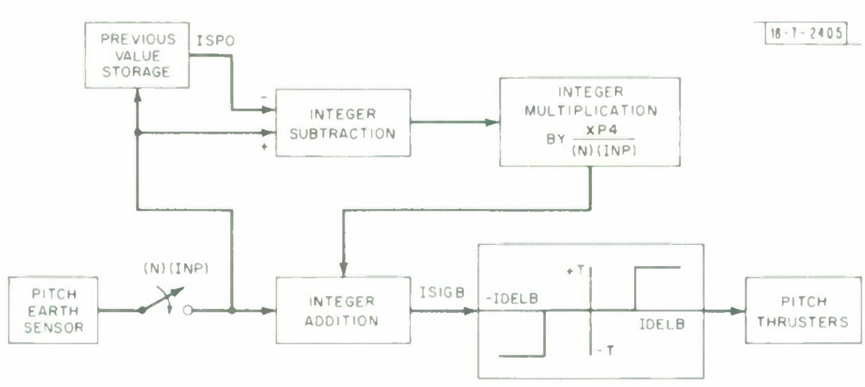

Fig. 4. Mode P4: Back-up mode.

system with the thrusters providing the control torque source. This mode provides a partial back-up to the mode P2 control law. It is assumed that the momentum wheel is running at constant speed in this mode. The control function ISIGB is given by the relation

$$\mathrm{ISIGB} = \frac{(\mathrm{XP4})}{\mathrm{N(INP)}} [\mathrm{ISP} - \mathrm{ISPO}] + \mathrm{ISP} \quad , \tag{27}$$

where

XP4 = constant design parameter controlling the damping in the system

ISPO = value of ISP at the previous sampling interval

INP = integer number of sampling intervals between update times; i.e., relation (27) is updated at intervals of N(INP) seconds.

The function ISIGB is nulled about a threshold value IDELB, with the control law:

$$\begin{aligned} &\text{If} \quad \mathrm{ISIGB} > \mathrm{IDELB} \quad , \quad \mathrm{ISEQU}_1 = 5 \\ &\text{If} \quad \mathrm{ISIGB} < -\mathrm{IDELB} \quad , \quad \mathrm{ISEQU}_1 = 1 \\ &\text{If} \quad |\mathrm{ISIGB}| \leqslant \mathrm{IDELB} \quad , \quad \mathrm{ISEQU}_1 = 0 \quad , \end{aligned} \tag{28}$$

where

IDELB = quantized threshold value of mode P4 pitch axis pointing accuracy

ISEQU_1 = logic input to thruster control matrix for torque about pitch axis (0, 1, 5 commands designate zero, plus and minus torque, respectively).

6. Mode P5

This mode is called the "gyro" control mode. It is a tentative mode and has not been simulated in detail yet. A preliminary coarse simulation of the gyro mode as envisioned for use in LES-8/9 is included here. The rate integrating gyro output ISAMP is assumed proportional to the integral of pitch axis rate, i.e.,

$$\mathrm{ISAMP(IS)} = \frac{1}{\mathrm{DELTAR}} \int_{t_o}^{t} Y_6 \, dt \quad . \tag{29}$$

To be compatible with the earth sensor interface the gyro output is quantized with the same resolution and same sampling time N as the earth sensors. This mode simply replaces the earth sensor with a rate gyro output. Detailed simulation of the gyro dynamics may be included at a later time if necessary.

B. Pitch Axis Coarse Momentum Control

In addition to the above modes, there is a coarse momentum expulsion limit control used when the system is in modes P2 or P3. This control is used to maintain the angular momentum stored in the wheel within ±10 percent of its nominal bias value. The coarse control law is a binary "on-off" hysteresis switch

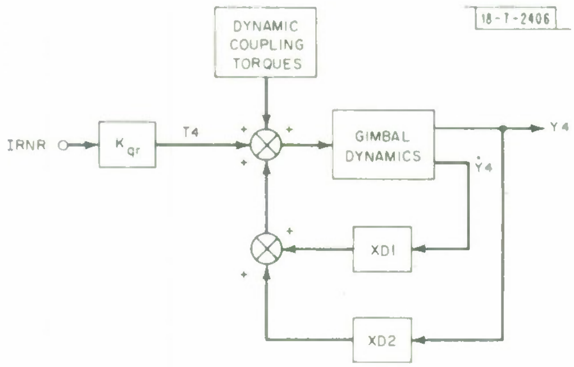

Fig. 5. Mode R1: Damping mode.

$$K_{qr} = (K_d K_t K_A) \left(\frac{\text{ft-lb}}{\text{bit}}\right)$$

where

$$K_d = \text{gain of D/A converter} \left(\frac{\text{volts}}{\text{bit}}\right)$$

$$K_t = \text{gain of gimbal torque motor} \left(\frac{\text{ft-lb}}{\text{A}}\right)$$

$$K_A = \text{gain of gimbal torque amplifier} \left(\frac{\text{A}}{\text{volt}}\right)$$

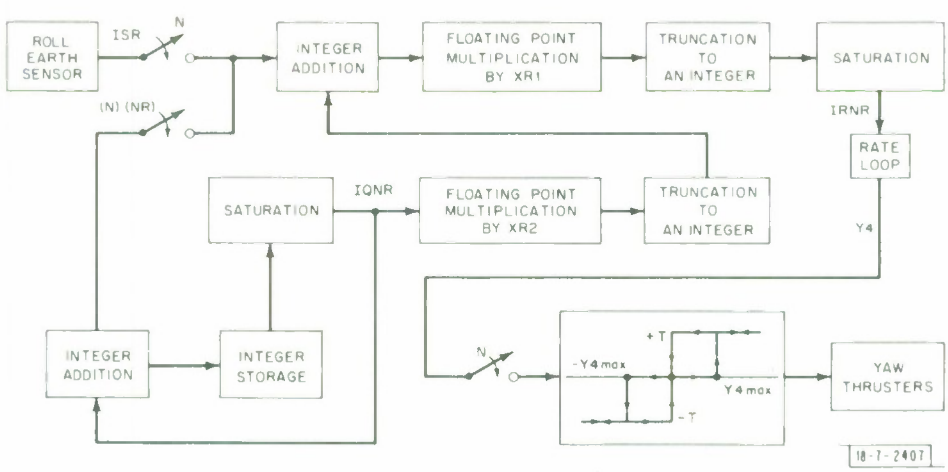

Fig. 6. Mode R2: Normal roll control.

If $|Y_5 - Y_{50}| < |Y_{50} - Y_{5\,min}|$, $ISEQU_1 = 0$

If $Y_5 > Y_{5\,max}$, $ISEQU_1 = 5$ until $(Y_5 - Y_{50}) \leqslant 0$

If $Y_5 < Y_5$ min , $ISEQU_1 = 1$ until $(Y_5 - Y_{50}) \geqslant 0$, (30)

where

$ISEQU_1$ = thruster logic input for pitch torque command

Y_{50} = nominal bias momentum value

$Y_{5\,min}$ = minimum bias momentum allowable before dumping

$Y_{5\,max}$ = maximum bias momentum allowable before dumping.

C. Roll Axis Control Modes

There are 4 operating modes available for controlling the satellite attitude about the roll axis. Three modes utilize momentum exchange capability of the gimbal control system and one mode uses the pulsed plasma thruster for direct momentum expulsion attitude control. The control laws utilized in these 4 modes are listed below.

1. Mode R1

This mode, shown in Fig. 5, is termed the "damping mode" and consists primarily of a single analog rate loop used to generate heavy damping of gimbal angle rates. In Refs. 2, 4 it was shown that the existence of this loop caused active nutation damping at all times and inertially stabilized the large angular momentum vector stored in the momentum wheel. In this mode the gimbal damping coefficient XD_1 and spring constant XD_2 are included in Eq. (1) as fixed control laws and the gimbal input IRNR = 0.

2. Mode R2

This mode, shown in Fig. 6, is termed the "normal" roll control mode. The input IRNR to the gimbal control system is a function of the earth sensor output ISR. The control law for mode R2 uses momentum exchange between the gimbal control system and satellite body to control the roll axis pointing. The gimbal torque motor provides the necessary reaction control torque to move the satellite body about the momentum wheel gimbal axis (roll axis). In addition, a coarse momentum expulsion limit control law is used to regulate maximum gimbal angle magnitude within fixed upper and lower bounds to prevent gimbal saturation and excessive yaw axis attitude[4] errors. The mode R2 control equations are given below.

$$IRNR = [ISR + (IQNR)\,XR2]\,(XR1) \quad . \tag{31}$$

This relation is updated at intervals of N seconds when the value of ISR is updated. The constants XR1, XR2 are parameters whose values are adjusted to yield optimum roll pointing performance in mode R2.

$$IQNR \triangleq \sum_{N(NR)} ISR \quad , \quad NR = \text{integer} \quad , \quad N = 4\text{ sec} \tag{32}$$

Relation (32) is updated at intervals of N(NR) seconds. The parameter NR is adjusted to optimize pointing performance in cooperation with XR1, XR2. The values of IQNR and IRNR are limited to the bounds

$$\text{QNRMIN} < \text{IQNR} < \text{QNRMAX}$$

$$|\text{IRNR}| \leqslant \text{RMAX} \quad , \tag{33}$$

to keep the gimbal control motor from saturating and thus provide no damping.

3. Mode R3

Mode R3, shown in Fig. 7, is termed the "gimbal control" mode. In this mode the control laws are identical to those used in mode R2 except that the earth sensor output ISR is replaced by the quantized gimbal angle sensor output Y_4.

$$\text{ISR} = -Y_4/\text{DELTAR} \quad . \tag{34}$$

This mode can be used to independently control the satellite orientation about the roll axis as, for instance, in an acquisition search scan, etc. This mode is a momentum exchange system also and relies upon gimbal control motor reaction torque.

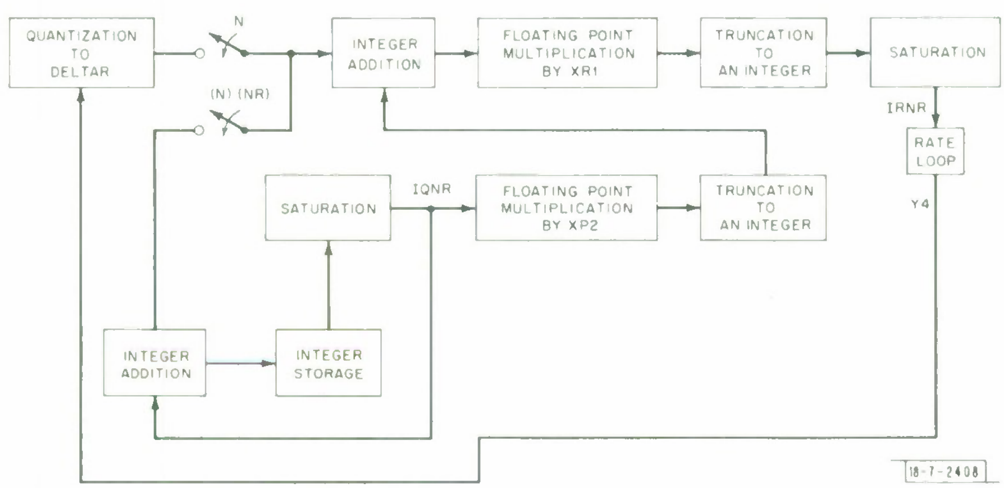

Fig. 7. Mode R3: Gimbal control.

4. Mode R4

This mode is termed the roll axis "back-up" mode (see Fig. 8). The control system consists of a binary "on-off" hysteresis control function whose sign controls the satellite attitude control thrusters. This system is a momentum expulsion system whose primary function is to provide a back-up capability to mode R2. This mode depends upon the momentum storage

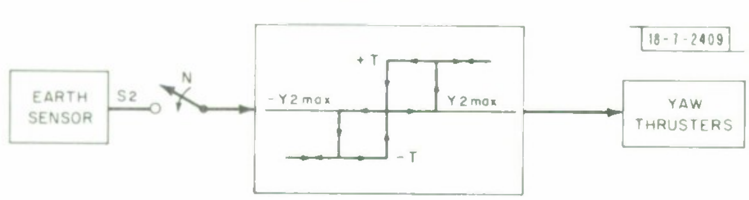

Fig. 8. Mode R4: Back-up mode.

capability of the wheel due to the steady state precession equations relating roll axis motion with torque about the yaw axis. The control law for this mode uses the roll axis earth sensor input S_2 to activate the yaw thrusters.

$$\text{If} \quad |S_2| < Y_{2\,max} \quad , \quad ISEQU_3 = 0$$

$$\text{If} \quad S_2 \geqslant Y_{2\,max} \quad , \quad ISEQU_3 = 3 \text{ until } S_2 \leqslant 0$$

$$\text{If} \quad S_2 < -Y_{2\,max} \quad , \quad ISEQU_3 = 7 \text{ until } S_2 \geqslant 0 \quad , \qquad (35)$$

where

$ISEQU_3$ = thruster logic input for yaw torque command

$Y_{2\,max}$ = quantized threshold value of mode R4 roll axis pointing accuracy.

D. Roll Axis Coarse Momentum Control

In addition to the above roll control modes, there is a coarse momentum expulsion limit control used when the system is in mode R2. This control is used to maintain the satellite angular momentum vector orientation perpendicular to the orbit plane within a given threshold value. The input to this control law is the gimbal angle Y_4. This control law maintains $|Y_4|$ within a threshold region at all times so that yaw attitude error is held to an acceptable value over the entire orbit. See Ref. 4 for a description of how Y_4 couples into Y_3 when the satellite is in mode R2. The control law for coarse momentum control is a binary "on-off" hysteresis switch as follows:

$$\text{If} \quad |Y_4| < Y_{4\,max} \quad , \quad ISEQU_3 = 0 \qquad (36a)$$

$$\text{If} \quad Y_4 > Y_{4\,max} \quad , \quad ISEQU_3 = 3 \text{ until } Y_4 \leqslant 0 \qquad (36b)$$

$$\text{If} \quad Y_4 < -Y_{4\,max} \quad , \quad ISEQU_3 = 7 \text{ until } Y_4 \geqslant 0 \quad , \qquad (36c)$$

where

$Y_{4\,max}$ = angular threshold value of mode R2 yaw axis pointing accuracy.

Note that relations (36b) or (36c) hold until the value of Y_4 reaches zero or changes sign and then control is shifted to relation (36a).

IV. DESCRIPTION OF SIMULATION

In this section, the subroutines which are used to perform the simulation are briefly described and flow diagrams of the new subroutines are given. The overall simulation is performed by 13 subroutines. These are broken into three basic groups according to their functions.

(a) Subroutine linkage and integration subroutines

Main Program
RK (Runge-Kutta)[1]
AM (Adams-Moulton)[1]

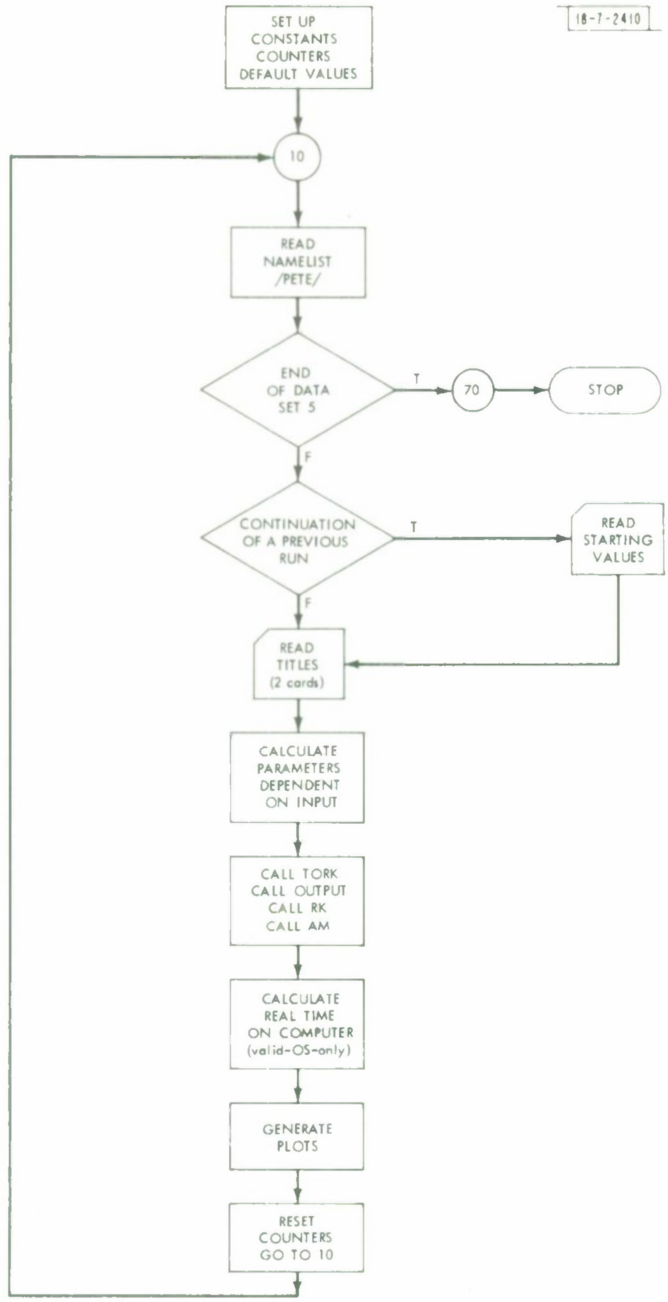

Fig. 9. Flow diagram of main program.

(b) Satellite dynamics, kinematics, sensors and control law subroutines

DERIV

DIGIT

TORK

TORQUE

UNLOAD

(c) Output format and plotting subroutines

GR 1

GR 2

OUTPUT

FRAME V

PRINT V

A. Main Program of the Simulation

The main routine sets up the initial conditions, reads each data set in turn, calls the simulation routines, and plots the results (see Fig. 9). There are several routines called in various sections of the program which are in the Lincoln Library. They are DUMPV, TIMHR,[5] REREAD, STOIDV, FRAMEV, PRINTV, PLTND.

The input is read in via NAMELIST/PETE/. All variables are computed in standard engineering units, degrees, degrees/second, rpm for the wheel speed. Those variables in the input list whose units have to be altered for the simulations are stored as they are initialized and new variables hold the converted values. The variable names differ by 0 (zero) as the final character of the variable name. Following the namelist, two title cards are read in. The program stores all 80 columns of the first card and the first 48 columns of the second card. Successive data sets may be added for each simulation desired. The program may be terminated in either of two ways: (1) the absence of data to be read in terminates the program normally, (2) the namelist may be read in with the logical variable FIN set to TRUE : FIN = T.

The program was originally written to produce punched output with the intention of continuing a particular simulation. Those write statements in subroutine AM on logical unit 7 are currently commented out. To read these cards back in, the user should initial all variables in the namelist as they originally were with the exception of the logical variable T00 : T00 = T. The punched cards follow the namelist and precede the titles in the input stream. When this method of initialization is used, the array Y0 has units used by the program rather than engineering units. For a successive data set, all of Y0 should be initialized.

To perform the requested simulation the program makes an initial call to TORK to initialize constants for the thrusting sequencing routine TORQUE and UNLOAD. These constants are used via the calls to the entry point THRUST in TORK. An initial call to OUTPUT stores the initial values of Y to be plotted. The sequence of calls to RK and AM perform the simulation. Control returns to MAIN when T exceeds TBOUND. The stored arrays PY1, PY2, . . . PYn are plotted via calls to GR1. NFR controls the number of frames the data are to be plotted on; PT is the array of times, PYi is the corresponding array of data. If PYi is constant throughout, no plot can be generated. IERR is returned nonzero. The calls to PRINTV label the last frame of each set of frames per array plotted. The first and last frame for each simulation are the input parameters and titles.

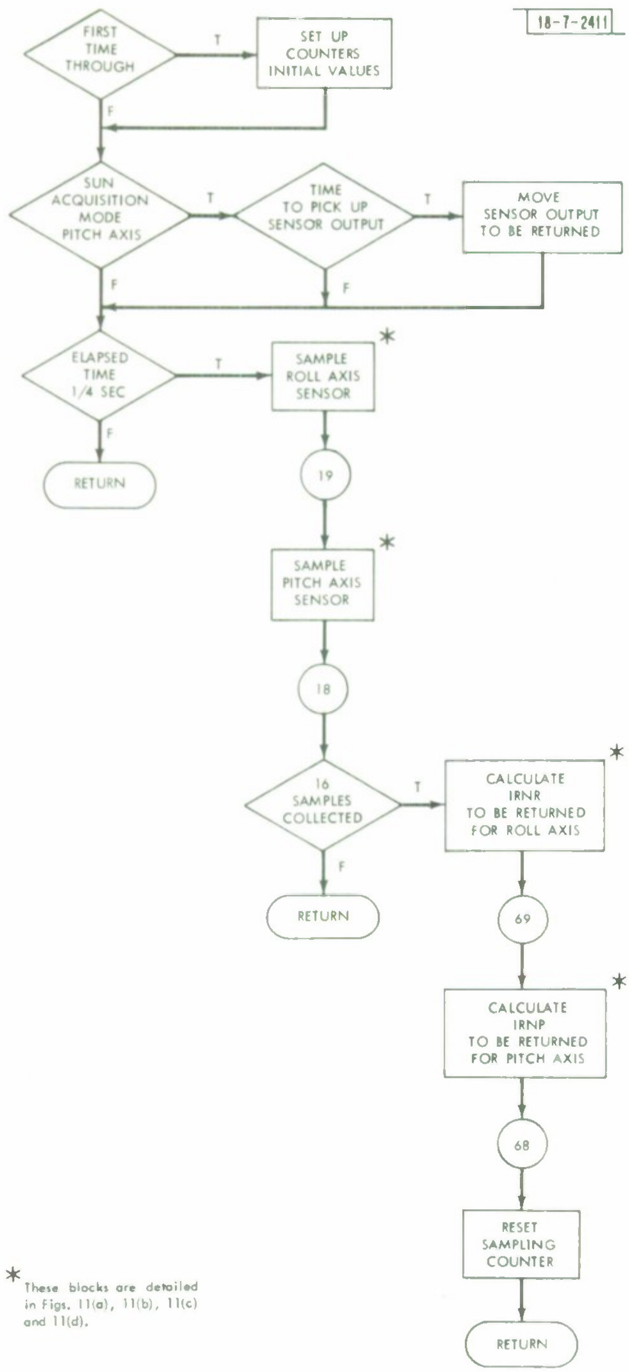

Fig. 10. Main flow of subroutine digit.

After resetting the appropriate parameters and counters, program control returns to the reader for another set of data for the next simulation. An empty reader terminates the program.

The current size of the program requires that simulations totaling 2000 seconds or less be run as Class C jobs under Lincoln's MVT, longer simulations are Class F.

B. Major Subroutines

1. DERIV Subroutine

Subroutine DERIV is called by RK and AM once for each step in the integration. The calling sequence is Y, YDOT. Y is the array of values coming in; YDOT is the array of values returned. In addition to calculating YDOT for each defined entry of the array, DERIV sets up the parameters needed by DIGIT, calls DIGIT, and updates the motor torque T_M using the results of DIGIT every T_P seconds.

The parameters IB and IBE are flags for the initial step of each simulation. They are zero until used once, then they are nonzero until a new simulation is begun.

The state variables Y, YDOT are dimensioned 20 in the calling programs. Currently, the program uses 12 of these 20 locations. To add to the existing system of differential equations, one need define YDOT(i) in DERIV, where $12 < i \leq 20$. The variable NEQ is initialized on BLOCK DATA and can be changed in the main program via NAMELIST. The parameter NEQ is the number of equations in the system being solved.

2. DIGIT Subroutine

Subroutine DIGIT contains the mathematical model of the actual digital control system on the pitch and roll axes. It receives from DERIV the current values of S(1), S(2), Y(4), Y(5), and Y(10). It returns to DERIV the values of the sensors for the pitch and roll axes, IRNP and IRNR.

The main flow of this routine is shown in Fig. 10 and is constructed with three major blocks. The first block is executed once at the beginning of each simulation. In this block all parameters necessary for the digital model and dependent on the particular input data set are calculated and all counters are initialized.

Parameter IB flags whether or not the first block has been executed. The second block stores the sensor output every quarter second [see Figs. 11(a) and 11(b)]. The third block averages the sensor output over 4 second intervals and uses these averages to calculate the returned values for the controlling system of equations.

In the first block, SIGMA is converted from degrees to least significant bits, and its value is adjusted for 4 second averaging effects by being multiplied by 4. Maximum and minimum saturation values are calculated for the integrator registers and these values are returned to DERIV. The variable T is the current time as calculated in RK and AM. TZERO is the initial time of each quarter second sampling period. IS is the incremented index to store the sampled information. IS is set to zero initially and after each set of 16 sensor samples.

If MODEP is 3, there is a delay in returning the pitch sensor output back to DERIV. The delay is controlled by TIMDEL, an input parameter. IP3 delays the program logic flow until the sensor output is generated before checking to see if elapsed time equals or exceeds the time delay. TP3 is the time that the sensor output was generated.

The second block of DIGIT stores the sensor output every quarter second to be eventually averaged (see Figs. 11(c) and 11(d). These are two principal controlling statements. The first

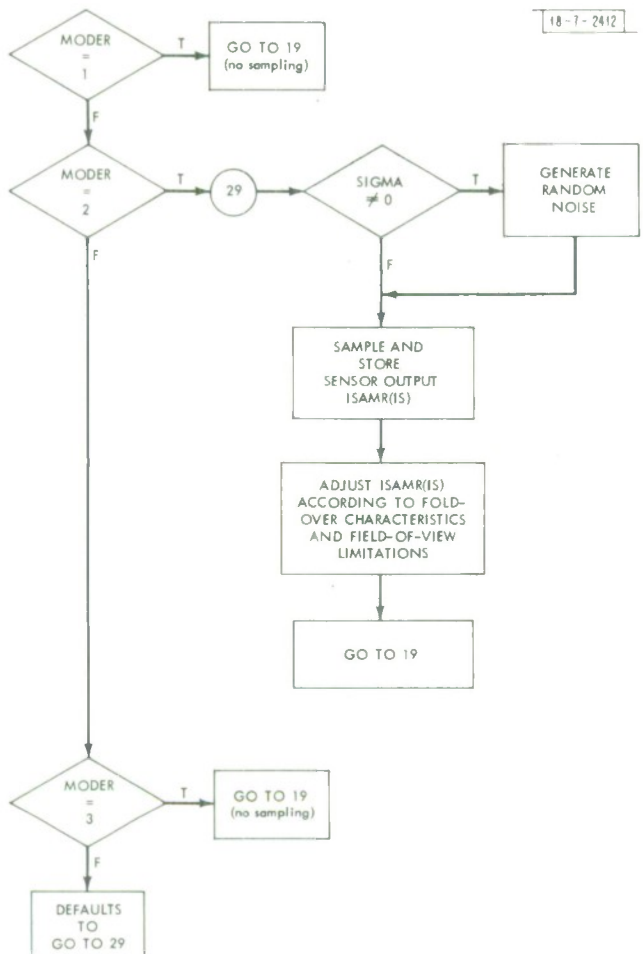

Fig. 11(a). Pitch axis sensor flow in digit.

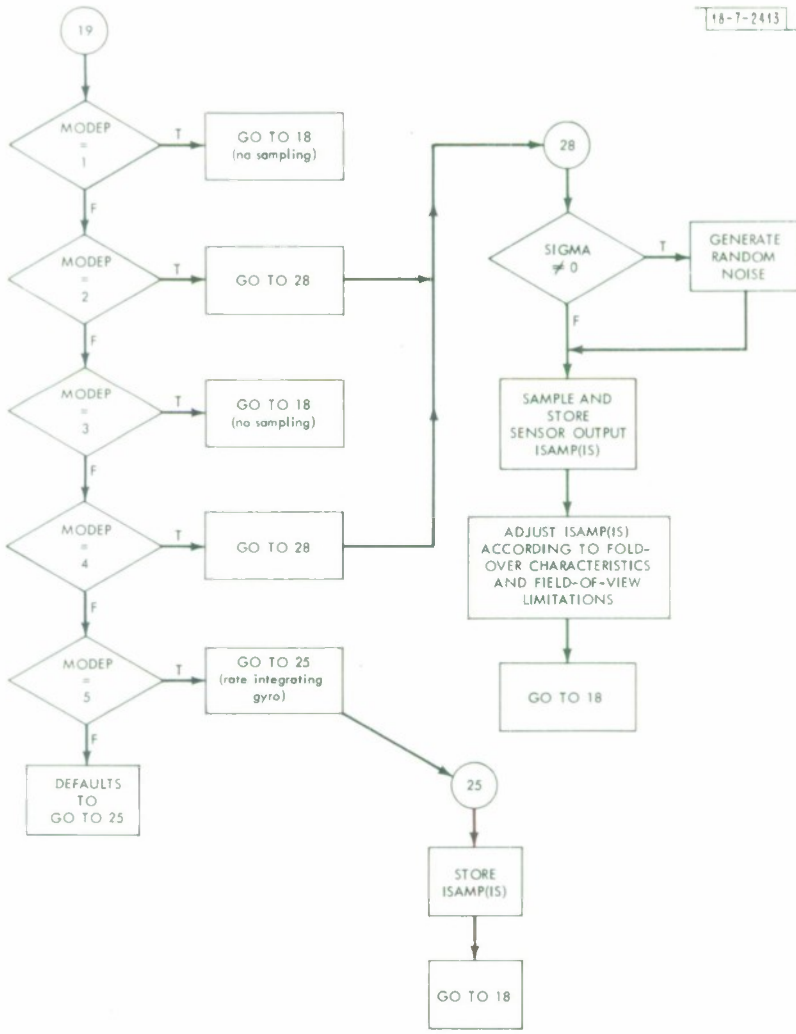

Fig. 11(b). Roll axis sensor flow in digit.

18-7-2414

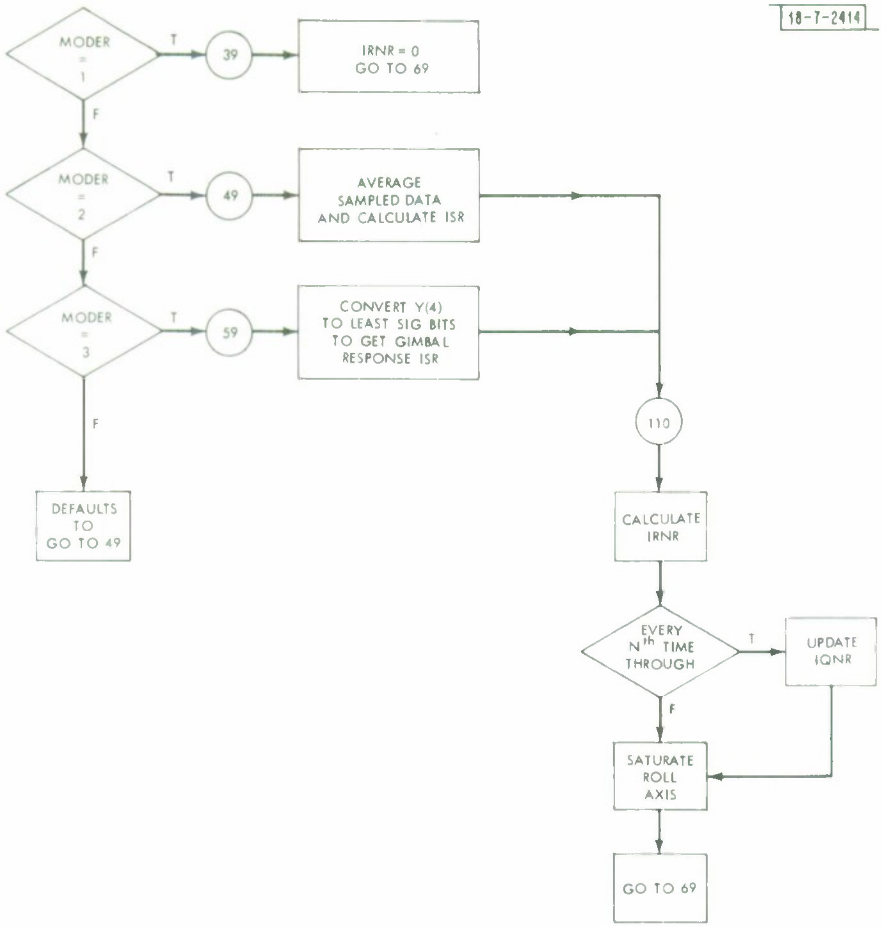

Fig. 11(c). Roll axis logic flow in digit.

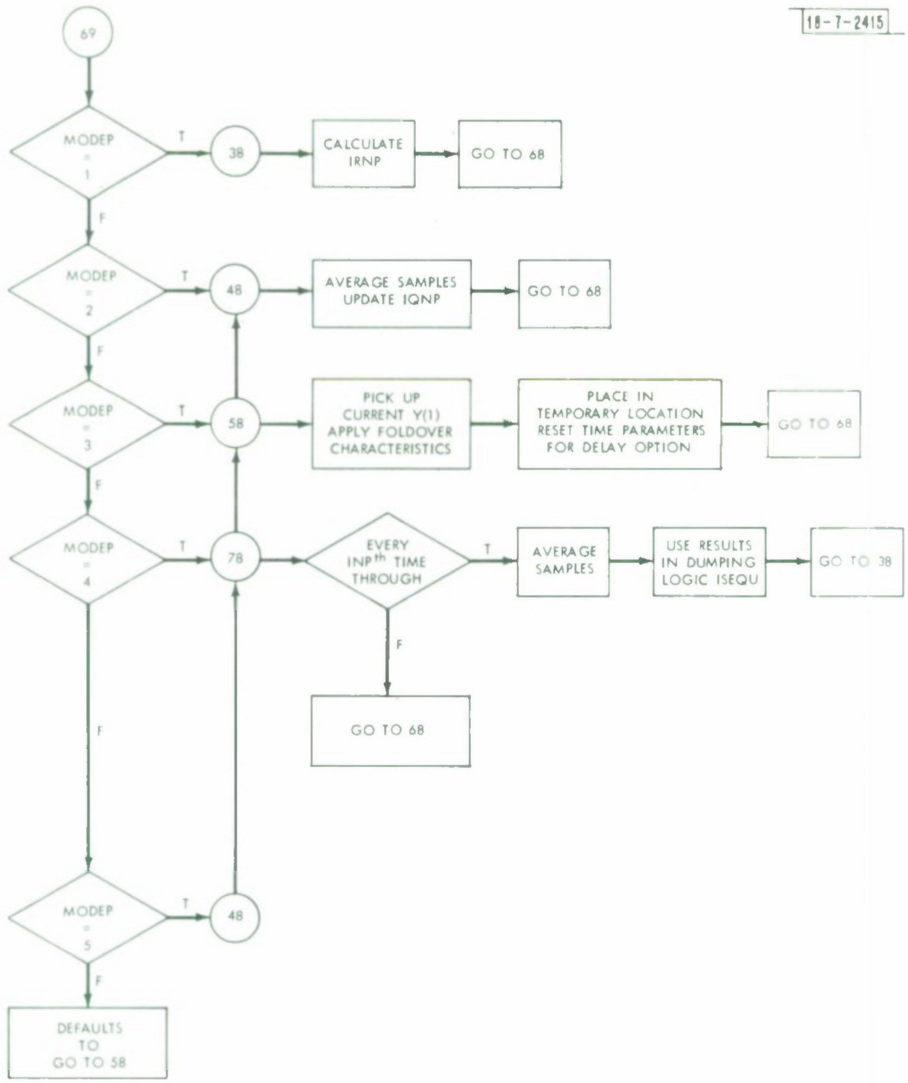

Fig. 11(d). Pitch axis logic flow in digit.

is a computer GO TO for the roll axis. The second is a computer GO TO for the pitch axis, i.e.,

GO TO (19, 29, 19), MODER.

When MODER = 1 or 3, the program logic flow goes to labeled statement 19. Likewise, when MODER = 2, program control branches to labeled statement 29.

	Value	Meaning
MODER	1	DAMPING MODE
	2	POINTING MODE
	3	GIMBAL CONTROL MODE
MODEP	1	CONSTANT SPEED
	2	POINTING MODE
	3	ACQUISITION MODE
	4	PITCH BACK-UP MODE
	5	RATE INTEGRATING GYRO MODE

To increase the number of modes for either roll or pitch axis, another labeled control section must be added and the label put in the list for the appropriate computed GO TO statement. In DERIV, regardless of which roll axis is used, control passes eventually to labeled statement 19, the computed GO TO for the pitch axis. The various pitch modes finally pass control to labeled statement 18 where TZERO is set, and IS is checked to see if 16 samples have been stored. If less than 16 samples are in storage, control returns to DERIV. Otherwise, control passes to the third block in DIGIT.

For the roll axis, no sampling takes place in modes 1 and 3. For mode 2, the values of Y(2), (S(2) in DERIV) are stored. The decimal fraction of Y(2) is truncated; the integrals remaining are then converted from radians to least significant bits. For a nonzero SIGMA, a random number from a Gaussian distribution is added to Y(2). The storage array is ISAMR. According to the value of Y(1), (S(1) in DERIV), limits of the field of view for the sensor are determined. If Y(1) is outside the field of view, FOV, a zero is stored. If the absolute value of ISAMR lies between 1024 and 2047, the difference between 2048 and ISAMR is stored with the signs of the original value. This process is referred to as the "fold over" sensor characteristic. When the roll axis sensor output is stored, the program then repeats the procedure for the pitch axis sensor.

For the pitch axis, no sampling takes place for modes 1 and 3. For mode 5, the variable Y(10) (the rate gyro output) is converted from radians to least significant bits and stored in ISAMP. For pitch modes 2 and 4 the current value of Y(1) plus any random noise is stored in ISAMP. The random noise is selected from a Gaussian distribution with standard deviation, SIGMA. For SIGMA equal to zero no noise is added. The program compares SIGMA ± EPS (where EPS is a small number) with zero, since testing on equality with real numbers is not always reliable. If Y(2) indicates that the sensor is out of the field of view, ISAMP is set to zero. The fold over sensor characteristic is identical for both the pitch and roll axis sensors.

After the pitch axis information is stored, TZERO is reset. IS is compared to see if 16 samples have been taken. The 16 samples parallel the actual 16 samples taken every 1/4 second for 4 seconds. A counter rather than absolute time is used because of the variety of time increments possible to be used by AM. Four seconds could mean 15 or 17 samples taken. If IS is less than 16, program control returns to DIGIT.

When IS equals 16, program control passes to the third block of DIGIT. The integer N is a continuous counter of the number of times sensor averaging has taken place. Thus, certain calculations can be done every i^{th} change in N by a comparison of N mod i with zero.

In the third block of DIGIT, the two principal control statements are computed GO TO's for roll and pitch axis. Any additional modes added in the second block must also be added to these GO TO's even though the averaging logic is the same as some previous mode. Whatever roll axis mode is used, control eventually passes to labeled statement 69. Statement 69 is the computed GO TO for the pitch axis. Each pitch mode eventually returns to labeled statement 68. Here the counter IS is reset to zero and control returns to DERIV.

For the roll axis, mode 1 returns a zero in the output IRNR. For mode 2, the 16 stored sensor values are averaged as ISR. For mode 3, Y(4) is converted from radians to degrees and then quantized and stored as ISR. The variables ISR and IQNR are used to calculate IRNR. IQNR is updated by ISR every NR^{th} time and is limited by QNRMAX and QNRMIN. The absolute value of IRNR is limited by RMAX and is returned to DIGIT. After this operation, control passes to the computed GO TO for the pitch axis.

For the pitch axis, mode 1 returns the initial wheel speed with appropriate unit manipulations via IINT as IRNP. For mode 2, the 16 samples in ISAMP are averaged and the result stored as ISP. INDEX is a counter for storage of time and sensor output to be plotted. IRNP is calculated using ISP and IQNP where IQNP is updated by ISP every NP^{th} time through the loop. IQNP is bounded by QNPMAX and QNPMIN. For mode 5, the output of sensor one, Y(1) is bounded by ± and converted from radians to least significant bits and stored as ISSP. For the time delay option TP3 is set to T, IP3 is set to 1.

For mode 4, control law calculations take place every INP^{th} time and this affects the values of ISEQU. ISEQU controls the sequence of the firing of the thrusters. In this mode IRNP is returned as in mode 1.

3. TORK, TORQUE, and UNLOAD Subroutines

The three subroutines TORK, TORQUE, and UNLOAD provide DERIV with the appropriate torques, L1, L2, L3, which simulate the firing of the thrusters. Subroutine TORK is called once per simulation from the main program to establish certain tables dependent on the physical location of the thrusters on the satellite body. Subroutine TORQUE is called once per integration step by both AM and RK. UNLOAD is called each time TORQUE is entered. If it is time to fire a thruster, TORQUE calls TORK via ENTRY THRUST. At this time torques L1, L2, L3 are updated for DERIV.

The thruster torques are passed through COMMON/CONTROL/ and they have different names in each routine.

Subroutine	Variable Name		
TORK	T(1)	T(2)	T(3)
TORQUE	L(1)	L(2)	L(3)
DERIV	L1	L2	L3

In each routine they are REAL*8 variables.

Subroutine TORK accepts as input the distance in inches of the thrusters from the origin of the satellite reference frame. TORK establishes a torque look-up table, TABLE, first by filling in the signs (+ or −) of the torque components for each thruster in a 12 × 3 array. It then computes the magnitudes of the torques and replaces each entry with the torque components

along the Z_1, Z_2, and Z_3 axes. THETA is the angle of thrust with respect to Z_2Z_1 plane and PHI is the angle of thrust with respect to Z_2Z_3 plane. FORCE is the average force per firing pulse for each thruster. The ENTRY THRUST routine adds the component torques of each of the thrusters fired to the total components stored in COMMON/CONTROL/.

Subroutine TORQUE contains the logic to keep track of which thruster is being fired, along with the length of firing time, and determines the sequence of the firing. Two arrays initialized in BLOCK DATA are critical to the functioning of TORQUE.

ASEQ is an integer array of numbers from 0 to 8 which specifies the sequence of torquing activity. The program will specify thruster torque about each of the three satellite axes for 100 seconds. At the end of each 100 second interval, the program will automatically shift to the next axis indicated. The following table establishes a correspondence between ASEQ array values and common torque about the axes.

Torque	ASEQ Index
no torque	0
+21 torque	1
+22 torque	2
+23 torque	3
+22 stationkeeping	4
−21 torque	5
−22 torque	6
−23 torque	7
−22 stationkeeping	8

The thruster selection rules can be easily changed. The thruster selection rules are implemented by a thruster selection table initialized by BLOCK DATA. The thruster selection table is kept in the labeled common block, SELECT. SELECT is an 8 × 4 integer array. The first index specifies the desired action according to the previous table while the second index specifies the selected thrusters. The thrusters and the indices used to specify them are listed in the following table for reference.

Thruster	Select Index
A_1	1
A_2	2
B_1	3
B_2	4
C_1	5
C_2	6
D_1	7
D_2	8
E_1	9
E_2	10
F_1	11
F_2	12

This table is used for the selection of up to four thrusters to provide torque about each axis.

ASEQU is passed in common/SEQU/ and SELECT in common/TABLES/. ATIME and ALIMIT are the current length of time of firing and maximum allowable time of firing about the current axis. TTIME and TLIMIT are the current length of time of firing and maximum allowable time of firing with the current thruster. APOINT specifies the current axis. TCUR specifies the current thruster. The call to THRUST is made with the appropriate value of IFIRE taken from an appropriate place in SELECT. With the three forces updated, control returns to DERIV.

At the beginning of TORQUE is a call to subroutine UNLOAD (see Fig. 12). The values of ASEQU are updated depending on the values of MODER and MODEP (ASEQU is called ISEQU in UNLOAD). First the roll axis modes are examined, then the pitch axis modes. If the roll axis mode is 1 or 3, there is a choice of a back-up mode, MODE = 2. If there is no back-up on these modes, UNLOAD does nothing. If MODE = 2, ISEQU(3) is changed to 0 or 3 or 7 depending on the previous value of ISEQU(3), the current value of Y(4) and the maximum limit on Y(4)(YMAX). In the back-up mode for the roll axis, ISEQU(3) becomes 0, 3, or 7 depending on the previous value of ISEQU(3), the current value of Y(2) and the limiting Y2MAX. This is referred to as the gimbal dump logic.

For the pitch axis in modes 1 and 2 the value of ISEQU(1) is changed to 0, 1, or 5 depending on the previous value of ISEQU(1), the current value of Y(5), the upper and lower limits, Y5MAX and Y5MIN, and the initial wheel speed Y050. This is referred to as the wheel dump logic.

C. Output Format and Plotting Subroutines

The printed and optional punched output is produced in subroutine AM. The plotted output is stored by OUTPUT and plotted by GR1 and GR2. OUTPUT is called periodically by AM, every 2 seconds. To alter the frequency, MPT should be changed in BLOCKDATA and AM. The frequency of points plotted is related to the initial step size H0.

$$\text{frequency} = \frac{256 * \text{H0}}{\text{MPT}} \quad .$$

Printed output is produced every TBOUND-T0/10 seconds, to yield 10 sets total per simulation. Punched output is produced at the end of each simulation. The punched output is in Z format as it is intended for machine use only. The punched output does not include the current sensor information from DIGIT.

Subroutine GR1 is called from the main program once for each variable to be plotted. The calling sequence is NFR, X, Y. NFR is the number of frames to spread the data over; X is the array of horizontal values; Y is the array of vertical values. GR1 calls GR2 passing necessary parameters via COMMON/GR/. GR2 calls the SC4020 routines necessary to produce a grid, points, and lines connecting the points. If all the Y values are constant, no grid can be drawn, as the scale on each grid is dependent on the range of the Y values. In case no grid can be drawn, a flag, IERR, is returned non-negative. No further processing is performed in GR2 and the flag is returned to the main program via common/SKIP/ so that the printing of the titles is skipped also.

If it is desired to add the plotting of another variable, the variable should be stored in an array in subroutine OUTPUT. The array should be dimensioned 5001, should be REAL * 4, and should be in common /OTPT/ in both OUTPUT and in MAIN. In MAIN two additional cards are needed, one call to GR1 and one conditional call to PRINTV to produce a title on the plot. Deleting a frame of output is similar.

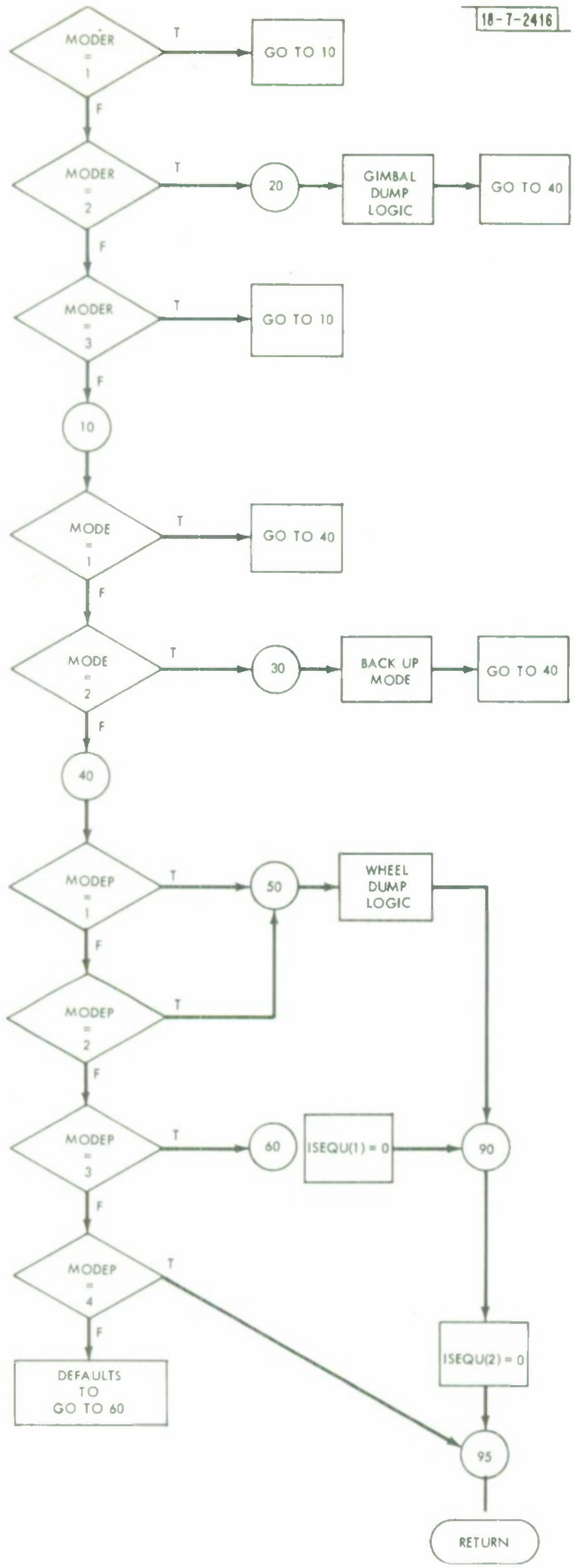

Fig. 12. Flow diagram of subroutine unload.

V. USE OF THE PROGRAM

The required program input variables along with typical values used and the output data format are described in this section. For each computer run input data are printed out at the beginning of the run to identify the parameter values used. The input variables are listed below in the same order as required in the program.

A. Input Variables

H0 = initial computation step size (seconds)

TBOUND = final computation time (seconds)

T0 = initial computation time (seconds)

Y0 = initial conditions of variables Y1 through Y12, respectively, at t = T0

NFR = integer number of frames of plotted output per simulation

FIN = F or T, logical false or true which stops program

T00 = F or T, logical false or true which is used to signify continuation of a run

GLB = greatest lower bound limit on integration error over one step (10^{-7})

LUB = least upper bound limit on integration error for one step (10^{-5})

XJ = principal inertias of satellite and reaction wheel rotor (XJ1, XJ2, XJ3, XJ4)

Z2 = thruster location distance from satellite center of gravity on coordinate Z2 (inches), one dimension

Z1 = thruster location distances from satellite center of gravity on coordinate Z1 (inches), two dimensions

ASEQU = three integers representing the initial values of thruster firing command about pitch, roll, yaw axes, respectively

XP1 = pitch axis proportional gain parameter (integer)

XP2 = pitch axis integral gain parameter

XD1 = nutation damping coefficient parameter

XD2 = gimbal flexible spring constant

XD3 = not being used (spare parameter)

XR1 = roll axis proportional gain parameter

XR2 = roll axis integral gain parameter

MODE = (1 or 2) roll axis back-up enable switch indicating normal or back-up control mode operation

SIGMA = RMS value of random noise output of earth sensor (deg)

NR = roll axis integral gain sampling ratio (integer)

MODER = integer representing desired roll axis operating mode (1-4)

MODEP = integer representing desired pitch axis operating mode (1-5)

MMODEP = integer number input to wheel speed control in pitch acquisition mode

IDEL = integer number representing deadband in pitch acquisition mode (bits)

TIMDEL = time delay in updating pitch acquisition mode logic after each sensor sample (seconds)

NP = pitch axis integral gain sampling ratio (integer)

M2 = gimbal misalignment angle about the Z_2 axis (radians)

M3 = gimbal misalignment angle about the Z_3 axis (radians)

XP3 = pitch axis derived rate gain parameter in acquisition mode (integer)

LSB = wheel speed period resolution (bits/second)

TD1 = external torque about pitch axis (ft-lbs)

TD3 = external torque about yaw axis (ft-lbs)

Y4MAX0 = maximum gimbal angle allowable in roll axis normal mode before momentum dumping (degrees)

Y5MIN0 = minimum wheel speed allowable before momentum dumping starts (rpm)

Y5MAX0 = maximum wheel speed allowable before momentum dumping starts (rpm)

Y2MAX0 = deadband in roll axis back-up mode (degrees)

Y5N0 = nominal bias wheel speed (rpm)

XP4 = pitch axis derived rate gain parameter in back-up mode (integer)

IDELB = integer number representing deadband in pitch back-up mode (bits)

INP = sampling ratio of pitch back-up logic to sensor sampling (integer)

Typical values for the above defined variables for simulation of the LES-8/9 current configuration are given below:

H0 = 0.25 sec, TBOUND = 300 $\rightarrow$ 1200 sec, T0 = 0.0

Y0 = (initial conditions for all state variables), Y_{10} = initial pitch axis error (degrees)

Y_{20} = initial roll axis error (degrees), Y_{30} = initial yaw axis error (degrees)

Y_{40} = initial gimbal angle (degrees), Y_{50} = initial wheel speed (rpm)

Y_{60} = initial pitch axis body rate (deg/sec), Y_{70} = initial roll axis body rate (deg/sec)

Y_{80} = initial yaw axis body rate (deg/sec), Y_{90} = not used = 0

Y_{100} = initial value of integral of Y_1, Y_{110} = initial ITAE value for pitch error

Y_{120} = initial ITAE value for roll error, NFR = 1, FIN = F,

T00 = T, GLB = 10^{-7}, LUB = 10^{-5},

XJ = (values of inertias): XJ1 = 120 slug-ft^2, XJ2 = 130 slug-ft^2

XJ3 = 28 slug-ft^2, XJ4 = 0.065 slug-ft^2, Z2 = 20 inches,

Z1 = 14.75, 16.25 inches, ASEQU = 0, 0, 0

XP1 = 2, XP2 = 0.125, XD1 = 6.0 ft-lbs-sec/rad, XD2 = 0.3 ft-lbs/rad

XD3 = 0, XR1 = 0.25, XR2 = 0.5 MODE = 1, SIGMA = 0.03 degrees

NR = 2, MODER = 2, MODEP = 2, MMODEP = 200,000, IDEL = 24,

TIMDEL = 0, NP = 3, M2 = 0, M3 = 0, XP3 = 6,

LSB = 500,000, TD1 = 0, TD3 = 0, Y4MAX0 = 0.1 degrees,

Y5MIN0 = 1000 rpm, Y5MAX0 = 1200 rpm, Y2MAX0 = 0.1 deg,

Y5N0 = 1100 rpm, XP4 = 200, IDELB = 20, INP = 5.

B. Output Data

Output data from the program described here is in the form of plots. The subroutines used to generate the output plots are described in Sec. IV-C. Typical transient response plots of the variables Y_1, Y_2, Y_5, versus times are illustrated in Figs. 13, 14, 15, respectively. The input data for these runs are listed in Sec. V-A. The plot coordinates are scaled in common units for ease of analysis; deg, deg/sec, bits, rpm, seconds.

In addition to the plots, each computer run contains a torque table printed out at the top of the input variable listing. This table is the output from subroutine TORK and consists of a

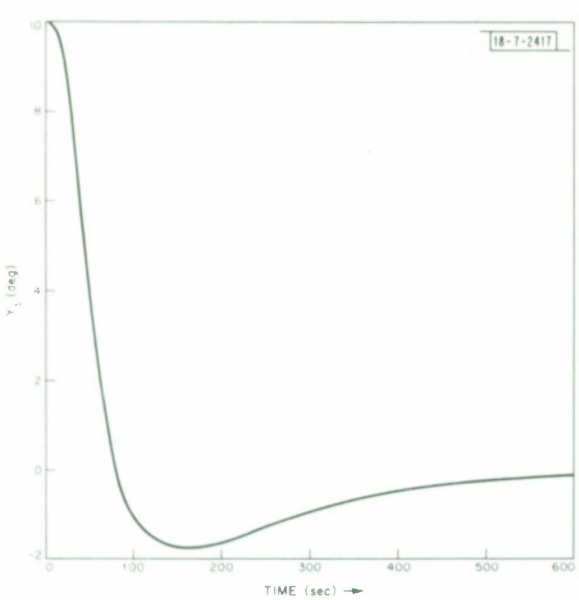

Fig. 13. Computer plot of pitch transient response.

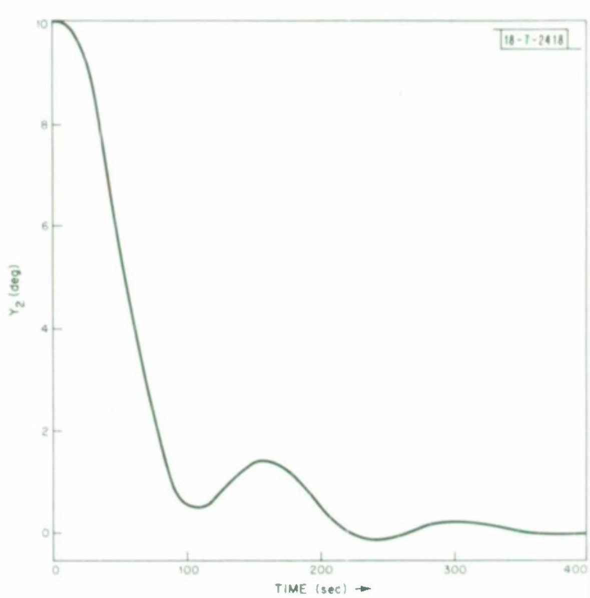

Fig. 14. Computer plot of roll transient response.

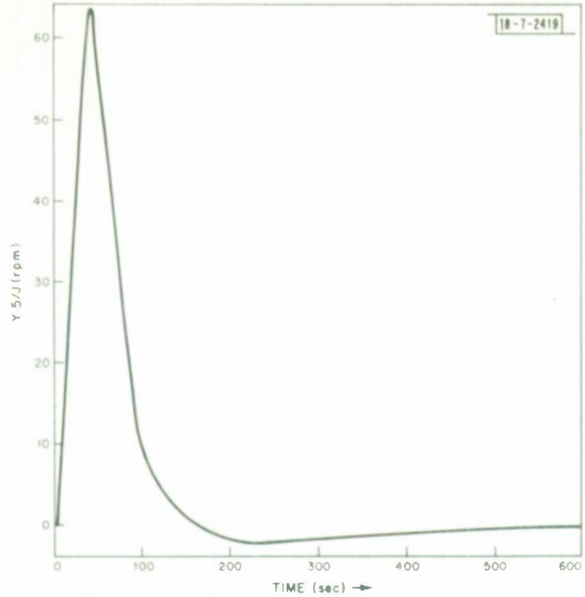

Fig. 15. Computer plot of reaction wheel speed response for Fig. 13.

3 × 12 matrix of torque components. This matrix consists of the components of torque impulse about each principal axis of the satellite resulting from each thruster on the satellite.

Also, included with each run is a printout of all variable values at a few discrete points in the solution time. These data are strictly for diagnostic purposes in monitoring the performance of the simulation. At the end of the output data is printed the final value of the performance index Y_{11} and Y_{12}. These indexes are useful for fine evaluation of control system transient response performance.

REFERENCES

1. R. W. Brockett and N. M. Brody, "Numerical Simulation of the Attitude Equations of a Satellite Containing Damping Rotors and a Reaction Wheel," Technical Note 1968-24, Lincoln Laboratory, M.I.T. (15 November 1968), DDC AD-679992.

2. J. U. Beusch and N. P. Smith, "Stable Equilibria of a Freely Spinning Satellite Containing a Wheel Mounted in an Active, Controlled Gimbal," AIEE Guidance, Control and Flight Mechanics Conference, August 1970.

3. A. Sabroff, R. Farrenhopf, A. Frew and M. Gran, "Investigation of the Acquisition Problem in Satellite Attitude Control," Technical Report AFFDL-TR-65-115, Air Force Systems Command, Wright-Patterson AFB (December 1965).

4. J. U. Beusch, F. W. Floyd, C. H. Much, V. J. Sferrino and N. P. Smith, "Three Axis Attitude Control of a Synchronous Communications Satellite," AIEE Third Communications Satellite Systems Conference, April 1970.

5. J. J. Fitzgerald, Accumulator, Issue No. 29, pp. 3-4, May 1967.

```
C
C
C SHORT TERM SIMULATION FOR LES-7                                        TEM00040
      IMPLICIT REAL*8 (A-H,O-Z)                                         TEM00050
      DIMENSION TITLE(16),Y(20),Y0(20),P(4),C(4),FY(20,5),XJ(4),LABEL(49TEM00060
     2),          Z1(2)                                                 TEM00070
      LOGICAL FIN,TOO,TERM                                              TEM00080
      INTEGER ASEQU(3), APOINT                                          TEM00090
      REAL*8 M2,M3
      REAL*4 PY1(5001),PY2(5001),PY3(5001),PY4(5001),PY5(5001),PT(5001),TEM00100
     2        TIME1, TIME2, ELTIME, ITIME                               TEM00110
      REAL*4 PY7(5001)
      REAL*4 XT,YT
      REAL*8 LUB, K                                                     TEM00120
     0COMMON /COEF/P,C,TBOUND,WR,MPT,NCT,NSTEP, NEQ                     TEM00130
     1        /OTPT/PT,PY1,PY2,PY3,PY4,PY5                              TEM00140
     2        /PARAM/XJ,XJ23,XJ31,XJ21,T1,T2,OMEGA0,XP1,XP2,XD1,XD2,XD3, TEM00150
     3        XR1,XR2                                                   TEM00160
     3        /ANGMOM/HSQ0                                              TEM00170
     4        /SEQU/ H, ATIME, TTIME, ASEQU, APOINT                     TEM00180
     5        /GR/ DUM(3), NPL                                          TEM00190
      COMMON /FREP/ Y050,INDEX
      COMMON /PLOT/ XT(5000),YT(5000)
      COMMON /ALIGN/ M2,M3
      COMMON /SKIP/ IERR
      COMMON/PITCH/XP4,IDELB,INP
      COMMON /ELLSN/ TIMDEL,TIM,SIGMA,IB,NP,MODER,NR,IRNP,IRNR,MODEP,
     2IDEL,MMODEP,IPNP,XP3,LSB
      COMMON /TEMP/ PY7
      COMMON /DISTUB/ TD1,TD3
      COMMON /DUMP/Y4MAX,Y5MIN,Y5MAX,Y2MAX,Y5N,MODE
      REAL*4 SP(5001,2)
      COMMON /BETTER/ S(2),SP
      DATA TOO,FIN/T,F/,Y0/20*0.D0/,GLB,LUB/1.D-7,1.D-5/               TEM00200
      DATA PI /3.141592653589793/, Z1, Z2 /14.75, 16.25, 20.0/         TEM00210
      NAMELIST /PETE/H0,TBOUND,T0,Y0,NFR,FIN,TOO,GLB,LUB,XJ,            TEM00220
     2                   Z2,Z1,ASEQU,XP1,XP2,XD1,XD2,XD3,XR1,XR2,MODE,
     3             SIGMA,NR,MODER,MODEP,MMODEP,IDEL,TIMDEL,NP,M2,M3,
     3             XP3,LSB,TD1,TD3,Y4MAX0,Y5MIN0,Y5MAX0,Y2MAX0,Y5N0
     4             ,XP4,IDELB,INP
C                                                                       TEM00240
C                                                                       TEM00250
      CALL STOIDV('ATTITUDE CONTROL SYSTEM PETE SMITH D-371  X5815',
     146,0)
      CALL DUMPV (264)
      CALL REREAD (8,300)                                               TEM00280
C                                                                       TEM00290
      TD1=0.0
      TD3=0.0
      H0=.125
      T0=0.0
      Y4MAX0=.1
      Y5MIN0=1000.
      Y5MAX0=1200.
      Y2MAX0=.1
      Y5N0=1100.
      NFR=1
      NEQ=12
      P(1)=-.37500                                                      TEM00300
```

```
      P(2)=37.D0/24.D0                                                  TEM00310
      P(3)=-59.D0/24.D0                                                 TEM00320
      P(4)=55.D0/24.D0                                                  TEM00330
      C(1)=1.D0/24.D0                                                   TEM00340
      C(2)=-5.D0/24.D0                                                  TEM00350
      C(3)=19.D0/24.D0                                                  TEM00360
      C(4)=.375D0                                                       TEM00370
      OMEGA0 = (2.0D+00)*PI/(24.0D+00*3.6D+03)                          TEM00380
C                                                                       TEM00390
C BEGINNING OF INPUT LOOP                                               TEM00400
C FOR PETE SMITH THE UNITS OF THE INPUT WERE CHANGED TO DEGREES
   10 READ (5,PETE,END=70)
C AND FOR THE WHEEL SPEED TO RPM
      INDEX=0
      RAD=57.29578
      DO 750 ISP=1,4
750   Y(ISP)=Y0(ISP)/RAD
      Y(5)=Y0(5)*.0068055
      Y050=Y(5)
      DO 751 IPS=6,NEQ
751   Y(IPS)=Y0(IPS)/RAD
      Y4MAX=Y4MAX0/RAD
      Y2MAX=Y2MAX0/RAD
      Y5MIN=Y5MIN0*.0068055
      Y5MAX=Y5MAX0*.0068055
      Y5N=Y5N0*.0068055
      IF (FIN) GO TO 70                                                 TEM00430
      IF (T00) GO TO 30                                                 TEM00440
      READ (5,20) T0,Y0                                                 TEM00450
   20 FORMAT (Z16)                                                      TEM00460
      DO 60 I=1, NEQ                                                    TEM00630
   60 Y(I)=Y0(I)                                                        TEM00640
      GO TO 40                                                          TEM00470
   30 T0=0.D0                                                           TEM00480
   40 READ (5,50) TITLE                                                 TEM00490
   50 FORMAT (10A8)                                                     TEM00500
C                                                                       TEM00510
C INITIAL ANGULAR MOMENTUM                                              TEM00520
      X=Y(5)*(Y(6)*Y(4)+Y(8))
      HSQ0=(XJ(1)*Y(6)+Y(5)*DCOS(Y(4)))**2+XJ(2)*Y(7)**2+(XJ(3)*Y(8)-
     1      Y(5)*DSIN(Y(4)))**2
      XJ23=XJ(2)-XJ(3)                                                  TEM00570
      XJ31=XJ(3)-XJ(1)                                                  TEM00580
      XJ21=XJ(2)-XJ(1)                                                  TEM00590
C                                                                       TEM00600
      H=H0                                                              TEM00610
      T=T0                                                              TEM00620
      TIM=T
      IPRN=0
      IB=0
      CALL TORK(Z1, Z2)                                                 TEM00650
      WR=(TBOUND-T0)/10.D0                                              TEM00660
      WRITE (6,PETE)                                                    TEM00670
      WRITE (6,61) TITLE                                                TEM00680
   61 FORMAT ('1',16A8)                                                 TEM00690
C                                                                       TEM00700
C BEGIN SIMULATION                                                      TEM00710
      TERM=.FALSE.                                                      TEM00720
      S(1)=0.0
```

```
      S(2)=0.0
      CALL OUTPUT(T,Y,TERM)                                              TEM00730
          CALL TIMHR(TIME1)                                              TEM00740
      CALL RK(H,T,Y,FY,TERM)                                             TEM00750
      CALL AM(H,T,Y,FY,GLB,LUB,TERM)                                     TEM00760
          CALL TIMHR(TIME2)                                              TEM00770
      ELTIME = TIME2 - TIME1                                             TEM00780
      ITIME = (ELTIME/NSTEP)*3600.0                                      TEM00790
      ELTIME = ELTIME*60.0                                               TEM00800
          WRITE(6,62) ITIME, ELTIME                                      TEM00810
   62     FORMAT(' ELAPSED TIME FOR ONE ITERATION =   ',F7.5,' SECONDS SIMTEM00820
     2ULATION TIME = ', F7.5, ' MINUTES' )                               TEM00830
      WRITE(6,64)Y(11)
      WRITE(6,65) Y(12)
64    FORMAT(//' PERFORMANCE INDICATEOR FOR PITCH AXISO ',G10.3)
   65 FORMAT(//' PERFORMANCE INDICATEOR FOR ROLL  AXISO ',G10.3)
C                                                                        TEM00840
C PLOT INTEGRAL CURVES.                                                  TEM00850
       WRITE(8,63)(YO(I),I=1,11),HO,TO,TBOUND,XJ
63    FORMAT('YO=',11(F9.4,' ,'),'     ','HO=',F6.1,',  T=',F6.1,' TO ',
     1F6.1,' SEC.  , J=',3(G10.3 ,',  '),'JA=',G10.3)
      READ (8,67) LABEL                                                  TEM00900
   67 FORMAT (49A4)                                                      TEM00910
C                                                                        TEM00920
      CALL FRAMEV(0)                                                     TEM00930
      CALL PRINTV(128,TITLE,0,975)                                       TEM00940
      CALL PRINTV(124,LABEL,0,895)
      CALL PRINTV(-14,'INITIAL VALUES',500,1010)
      CALL PRINTV(72,LABEL(32),0,865)
      CALL GR1(NFR,PT,PY1)                                               TEM00960
      IF (IERR.EQ.0)CALL PRINTV(-11,'EPSLN1 VS T',0,1007)
      CALL GR1(NFR,PT,PY2)                                               TEM00990
      IF (IERR.EQ.0)CALL PRINTV(-11,'EPSLN2 VS T',0,1007)
      CALL GR1(NFR,PT,PY3)                                               TEM01020
      IF (IERR.EQ.0)CALL PRINTV(-11,'EPSLN3 VS T',0,1007)
      CALL GR1(NFR,PT,PY4)                                               TEM01050
      IF (IERR.EQ.0)CALL PRINTV(-11,' THETA VS T',0,1007)
      CALL GR1(NFR,PT,PY5)                                               TEM01080
      IF (IERR.EQ.0)CALL PRINTV(-14,'WHEEL-MOM VS T',0,1007)
      CALL GR1(NFR,PT,PY7)
      IF(IERR.EQ.0)CALL PRINTV(-4,'Y(6)',0,1007)
      CALL GR1(NFR,PT,SP(1,1))
      IF(IERR.EQ.0)CALL PRINTV(-13,'SENSOR 1 VS T',0,1007)
      CALL GR1(NFR,PT,SP(1,2))
      IF(IER.EQ.0) CALL PRINTV(-13,'SENSOR 2 VS T',0,1007)
      NPL=INDEX
      IF(MODEP.EQ.2)CALL GR1(NFR,XT  ,YT)
      CALL FRAMEV(0)                                                     TEM01100
      CALL PRINTV(128,TITLE,0,975)                                       TEM01110
      CALL PRINTV(124,LABEL,0,895)
      CALL PRINTV(72,LABEL(32),0,865)
C                                                                        TEM01130
      NPL=0                                                              TEM01140
      TTIME = 1.0D+03                                                    TEM01150
      ATIME = 1.0D+03                                                    TEM01160
      APOINT = 0                                                         TEM01170
      NSTEP = 0                                                          TEM01180
      GO TO 10                                                           TEM01190
   70 CONTINUE                                                           TEM01200

      CALL PLTND                                                         TEM01210
      STOP                                                               TEM01220
      END                                                                TEM01230
```

****

```
      SUBROUTINE DERIV(Y,YDOT)                                          TEM02620
C     1   2    3    4        5       6    7    8  9  10  11             TEM02630
C Y=EPS1,EPS2,EPS3,THETA,WHEEL-MOM,OM1,OM2,OM3,X1,X2,X3                TEM02640
C                                                                       TEM02650
      IMPLICIT REAL*8 (A-H,O-Z)                                         TEM02660
C TRQUES                                                                TEM02670
      REAL*8 M2,M3
      REAL*8 L1, L2, L3                                                 TEM02680
      DIMENSION Y(1),YDOT(1),XJ(4)                                      TEM02690
      COMMON /ALIGN/ M2,M3
      COMMON /ELLFN/ TIMDEL,TIM,SIGMA,IB,NP,MODER,NR,IRNP,IRNR,MODEP,
     2IDEL,MMODEP,IPNP,XP3,LSB
      COMMON /PARAM/XJ,XJ23,XJ31,XJ21,T1,T2,OMEGA0,XP1,XP2,XD1,XD2,XD3, TEM02700
     2 XR1,XR2                                                          TEM02710
     2          /CONTRL/ L1, L2, L3                                     TEM02720
      COMMON /DISTUB/ TD1,TD3
      REAL*4 SP(5001,2)
      COMMON /BETTER/ S(2),SP
C                                                                       TEM02730
C DEFINTION OF CONTROL VARIABLES                                        TEM02740
C  NAME       DESCRIPTION      SYSTEM       UNITS                       TEM02750
C  XP1            PROP             PITCH        SEC. D -1               TEM02760
C  XP2            INTEGRAL         PITCH        SEC. D-2                TEM02770
C  XD1            DERIVAIIVE       DAMPER       NONE                    TEM02780
C  XD2            PROP             DAMPER       SEC. D -1               TEM02790
C  XR1            PROP             ROLL         SEC D -1                TEM02810
C  XR2            INTEGRAL         ROLL         SEC D -2                TEM02820
C                                                                       TEM02830
      T1=.0647-.446D-3*Y(5)/XJ(4)
      T2=-.00107+.6627D-4*Y(5)/XJ(4)
      CY1=DCOS(Y(1))
      SY1=DSIN(Y(1))
      CY2=DCOS(Y(2))
      SY2=DSIN(Y(2))
      CY3=DCOS(Y(3))
      SY3=DSIN(Y(3))
      TY2=SY2/CY2
      S(1)=DATAN2(SY1*CY3+CY1*SY3*SY2, CY1*CY2)
      S(2)=DATAN2(SY2*CY1*CY3-SY1*SY3,CY1*CY2)
      TP=(6.283185D0)/(Y(5)/XJ(4)-Y(6)+Y(8)*Y(4))
      IPNP=IDINT(TP*DFLOAT(LSB))
      IF(IB.EQ.0)IBE=0
      CALL DIGIT (Y(4),Y(5),Y(10))
      IF(IBE.NE.0) GO TO 10
      IBE=10
      GO TO 11
10    IF((TIM-TLAST).LT.TAU) GO TO 15
11    IERROR=IPNP-IRNP
      TLAST=TIM
      TAU=TP
15    CONTINUE
      THEDOT=(-1.0D0/XD1)*(XD2*Y(4)+Y(5)*(Y(8)+(Y(4)+M2)*Y(6)
     1                               -M3*Y(7)*(Y(4)+M2))
     2                               -0.558D-04*DFLOAT(IRNR))
      HTHET = Y(5)*(THEDOT + Y(7))                                      TEM02870
C MOTOR TORQUE                                                          TEM02890
      TM=T1
      IF(IERROR.LE.0.0D0)TM=-T2
C FIRST ORDER EULER ANGLES                                              TEM02920
```

```
      YDOT(1)=(Y(6)*CY3-Y(7)*SY3)/CY2-OMEGAO
      YDOT(2)=Y(6)*SY3+Y(7)*CY3
      YDOT(3)=Y(8)-TY2*(Y(6)*CY3-Y(7)*SY3)
C ROLL AXIS                                                              TEM02960
      YDOT(4) = THEDOT                                                  TEM02970
C PITCH AXIS                                                             TEM02980
      YDOT(5) = TM                                                      TEM02990
C         RATES                                                          TEM03000
      H1=L1+TD1
      H3=L3+TD3
      YDOT(6)=(H1+XJ23*Y(7)*Y(8)+(Y(4)+M2)*HTHET-TM-M3*Y(5)*Y(8))/XJ(1)
      YDOT(7)=(L2+XJ31*Y(6)*Y(8)+M3*TM-Y(5)*((Y(4)+M2)*(Y(6)+M3*THEDOT)+
     1Y(8)))/XJ(2)
      YDOT(8)=(H3-XJ21*Y(6)*Y(7)+HTHET+(Y(4)+M2)*TM+M3*Y(5)*Y(6))/XJ(3)
C PITCH AXIS CONTROL LAW                                                 TEM03040
      YDOT(9)=Y(1)                                                      TEM03050
      YDOT(10)=Y(6)-OMEGAO
      YDOT(11)=DABS(Y(1))*TIM
      YDOT(12)=DABS(Y(2))*TIM
      RETURN                                                            TEM03080
      END                                                               TEM03090
                         *************************
```

```
      SUBROUTINE DIGIT (Y4,Y5,Y10)
      IMPLICIT REAL*8 (A-H,O-Z)                                         DIG00020
      DIMENSION ISAMP(16),ISAMR(16)
      INTEGER QNPMIN,QNRMIN,QNPMAX,QNRMAX                               DIG00040
      COMMON /ELLEN/TIMDEL,T,SIGMA1,IB,NP,MODER,NR,IRNP,IRNR,MODEP,IDEL,
     1MMODEP,IPNP,XP3,LSB
      COMMON/PARAM/XJ(4),DUM(6),XP1,XP2,XD1,DUMM(2),XR1,XR2
      REAL*4 SP(5001,2)
      COMMON /BETTER/ Y(2),SP
      REAL*4 XT,YT                                                      DIG00070
      COMMON /PLOT/ XT(5000),YT(5000)                                   DIG00080
      COMMON /FREQ/ YC50,INDEX                                          DIG00090
      COMMON/SEQU/DUM2(3),ISEQU(3)
      COMMON/PITCH/XP4,IDELB,INP
      COMMON /COEF/ZZX(8),TBOUND
      EQUIVALENCE (XJ4,DUM(4)),(ZP1,XP1)                                DIG00100
C                                                                       DIG00110
      DATA DELTAT/0.25D0/,DELTAD/0.011/,DELTAR/0.192D-3/,KRAN/156278590/DIG00120
      DATA EPS/1.D-12/,FOV/0.17453D0/                                   DIG00130
C COMMON /ELLEN/ APPEARS IN MAIN,AM,DERIV,RK                            DIG00140
C                                                                       DIG00150
C SUBROUTINE DIGIT                                                      DIG00160
C      SIMULATES DIGITAL CONTROL ON ROLL AXIS AND ON PITCH AXIS OF LES7  DIG00170
C                                                                       DIG00180
C LOOP PROCESSSED AT INITIAL CALL TO DIGIT FOR EACH SET OF INPUT DATA   DIG00190
      IF(IB.NE.0) GO TO 10                                              DIG00200
      TZERO=T                                                           DIG00210
      N=0                                                               DIG00220
      IS=0                                                              DIG00230
      ISP0=0
      IB=10                                                             DIG00240
      TWOPI=6.283185300
C CONVERT SIGMA TO LSB UNITS    DELTA =.011 DEG                         DIG00250
C CALCULATE PROBABILITY                                                 DIG00270
C SIGMA1 IS FOR A 4 SEC RMS
      SIGMA4=SIGMA1*4
      SIGMA=SIGMA4/DELTAD
      P=(SIGMA)**2/2.D4
      XLSB=DFLOAT(LSB)
      IINT=IDINT((6.283185D0*XJ(4)*XLSB)/(YC50*XP2))
      IQNP=IDINT((6.283185D0*XJ(4)*XLSB)/(Y5   *XP2))
      QNPMAX=IDINT(((6.823185D0*9.55D0* XLSB)/6.D2)/XP2)
      QNPMIN=IDINT(((6.283185D0*9.55D0* XLSB)/13.D2)/XP2)
      QNRMIN=IDINT(-2.0**10./(XR1*XR2))
      QNRMAX=IABS(QNRMIN)
      RMAX=2.D0**10.00                                                  DIG00330
      IQNR=0                                                            DIG00340
      IRNR=0                                                            DIG00370
      IRNP=IDINT((DFLOAT(IQNP))*(XP2))                                  DIG00380
      IP3=0                                                             DIG00390
      DELTS=3.1415926/4096.D0                                           DIG00400
      ISSP0=0                                                           DIG00410
      IRNPT=0                                                           DIG00430
      IRNPC=IPNP
      GO TO 10                                                          DIG00440
C                                                                       DIG00450
C  TEST FOR PICKING UP SENSOR OUTPUT IN SUN ACQUISITION MODE (PITCH AXISDIG00460
C                                                                       DIG00470
10    IF(MODEP.NE.3) GO TO 109                                          DIG00480
```

```
      IF(IP3.EQ.0)GO TO 109                                              DIG00490
      IF(T-TP3.LT.TIMDEL) GO TO 109                                      DIG00500
      IP3=0                                                              DIG00510
      IRNP=IRNPT                                                         DIG00520
C                                                                        DIG00530
C *** TEST TO SEE IF ELAPSED TIME IS 1/4 SEC ***                         DIG00540
C                                                                        DIG00550
109   IF(T-TZERO.LT.DELTAT) RETURN                                       DIG00560
      IS=IS+1                                                            DIG00570
C                                                                        DIG00580
C DIGITAL CONTROL LOOPS -- EVERY 1/4 SEC                                 DIG00590
C     1. ADJUSTABLE GAIN PARAMETERS                                      DIG00600
C     2  ADJUSTABLE INTEGRATOR SAMPLE PERIOD AND SATURATION LOOP         DIG00610
C     3  SENSOR FOLD-OVER CHARACTERISTICS                                DIG00620
C     4  SENSOR NOISE                                                    DIG00630
C     5  MODE SWITCH -- PITCH AXIS                                       DIG00640
C           MODEP =1 CONSTANT SPEED                                      DIG00650
C           MODEP =2 POINTING                                            DIG00660
C           MODEP =3 ACQUISITION                                         DIG00670
C     6  INTEGER ARITHMETIC                                              DIG00680
C     7  MODE SWITCH -- ROLL AXIS                                        DIG00690
C           MODER = 1 DAMPING                                            DIG00700
C           MODER = 2 POINTING                                           DIG00710
C           MODER = 3 GIMBAL CONTROL                                     DIG00720
C                                                                        DIG00730
C ******ROLL AXIS CONTROL SECTION ************                            DIG00740
C                                                                        DIG00750
      GO TO (19,20,19),MODER                                             DIG00760
20    RMSR=0.0                                                           DIG00770
C                                                                        DIG00780
C GENERATE RANDOM NUMBER                                                 DIG00790
C                                                                        DIG00800
      IF(SIGMA+EPS.GE.0.D0.AND.SIGMA-EPS.LE.0.D0) GO TO 129              DIG00810
      DO 119 I=1,100                                                     DIG00820
      IF(RAN2(KRAN).LE.P)RMSR=RMSR+10.
119   IF(RAN2(KRAN).LE.P) RMSR=RMSR-10.
C                                                                        DIG00850
C FOLD OVER SENSOR CHARACTERISTIC AND FIELD OF VIEW                      DIG00860
C                                                                        DIG00870
129   ISAMR(IS)=IDINT(Y(2)/DELTAR+RMSR)                                  DIG00880
      YMOD=DMOD(Y(1),TWOPI)
      IF(DABS(YMOD).GT.FOV)ISAMR(IS)=0                                   DIG00890
      IF(IABS(ISAMR(IS)).GE.1024.AND.IABS(ISAMR(IS)).LT.2048)            DIG00900
     1 ISAMR(IS)=(2048-IABS(ISAMR(IS)))*ISIGN(1,ISAMR(IS))               DIG00910
      IF(IABS(ISAMR(IS)).GE.2048)ISAMR(IS)=0                             DIG00920
C                                                                        DIG00930
C ****** PITCH AXIS CONTROL SECTION **********                           DIG00940
C                                                                        DIG00950
19    GO TO (18,28,18,28,25),MODEP
C  MODEP=5 RATE INTEGRATING GYRO SIMULATION FOR PITCH AXIS
25    ISAMP(IS)=IDINT(Y10/DELTAR)
      GO TO 18
C                                                                        DIG00970
C NOISE GENERATOR                                                        DIG00980
C     INPUT0 SIGMA   (USED TO CALCULATE P FIRST TIME THROUGH)            DIG00990
C     SPECIFY SENSOR RMS NOISE LEVEL IN DEGREES FOR ONE SCAN 0 SIG       DIG01000
C     OUTPUT0  RMS                                                       DIG01010
C     DIGITAL NUMBER WITH ZERO MEAN AND RMS IS APPROX. SIG FOR ADDITION  DIG01020
C     TO SAMPLED, QUANTIZED PITCH ERROR                                  DIG01030
```

```
C
28    RMS=0.0
      IF(SIGMA+EPS.GE.0.D0.AND. SIGMA-EPS.LE.0.D0) GO TO 12
C
C GENERATE RANDOM NUMBER
C
      DO 11 I=1,100
      IF(RAN2(KRAN).LE.P) RMS=RMS+10.
11    IF(RAN2(KRAN).LE.P) RMS=RMS-10.
12    YMOD=DMOD(Y(1),TWOPI)
      ISAMP(IS)=IDINT(YMOD/DELTAR+RMS)
C
C FOLD OVER SENSOR CHARACTERISITIC AND FIELD OF VIEW
C FIELD OF VIEW LIMITATION ON ROLL ERROR FOR PITCH SENSOR
      IF(DABS(Y(2)).GT.FOV)ISAMP(IS)=0
C FOLD OVER CHARACTERISTIC WITHIN FOV
      IF(IABS(ISAMP(IS)).GE.1024.AND.IABS(ISAMP(IS)).LT.2048)ISAMP(IS)=
     1(2048-IABS(ISAMP(IS)))*ISIGN(1,ISAMP(IS))
      IF(IABS(ISAMP(IS)).GE.2048) ISAMP(IS)=0
C
C
C ********** CHECK TO SEE IF ELAPSED TIME EQUALS 4 SEC. **********
C
18    TZERO=T
      IF(IS.LT.16) RETURN
      N=N+1
C
C *** ROLL AXIS ****
C
      GO TO (39,49,59),MODER
49    ISR=0
      DO 159 I=1,16
159   ISR=ISR+ISAMR(I)
      ISR=ISR/16
      GO TO 110
59    ISR=-IDINT(Y4/DELTAR)
110   IRNR=IDINT(DFLOAT(ISR+IDINT(DFLOAT(IQNR)*XR2))* XR1)
      IF(MOD(N,NR).EQ.0)IQNR=IQNR+ISR
      IF(IQNR.GT.QNRMAX)IQNR=QNRMAX
      IF(IQNR.LT.QNRMIN)IQNR=QNRMIN
C
C  SATURATION FOR ROLL AXIX  RNR
C
      RTEMP=DFLOAT(IRNR)
      IF(IABS(IRNR).GT.RMAX)IRNR=RMAX*DSIGN( 1.0D0,RTEMP)
      GO TO 69
39    IRNR=0.0
C
C **** PITCH AXIS CONTROL SECTION ****
C
69    GO TO (38,48,58,78,48),MODEP
58    YMOD=DMOD(Y(1),TWOPI)
      ISSP=IDINT(YMOD/DELTS)
      ISSP=-1*ISSP
      IAS=IABS(ISSP)
      IF((AS.GE.1024.AND.IAS.LT.2048) ISSP=1024*ISIGN(1,ISSP)
      IF((AS.GE.2048)ISSP=0
      IDAP=(ISSP-ISSP0)/4
      ISSP0=ISSP
```

```
      ISIG=IDAP*IDINT(XP3)+ISSP                                         DIG01600
      IF(IABS(ISIG).GT.IDEL) GO TO 56
      IRNPT=IRNPC
      GO TO 57
56    IRNPT=MMODEP*ISIGN(1,ISIG)
      IRNPC=IPNP
57    IP3=1
      TP3=T                                                             DIG01640
      GO TO 68                                                          DIG01650
C                                                                       DIG01660
C AVERAGING LOGIC                                                       DIG01670
C                                                                       DIG01680
48    ISP=0                                                             DIG01690
      DO 15 I=1,16                                                      DIG01700
15    ISP=ISP+ISAMP(I)                                                  DIG01710
      ISP=ISP/16*(-1)                                                   DIG01720
      INDEX=INDEX+1                                                     DIG01730
      XT(INDEX)=T                                                       DIG01740
      YT(INDEX)=ISP                                                     DIG01750
C                                                                       DIG01760
C INTEGRATOR SATURATION LOGIC - PITCH AND ROLL                          DIG01770
C THE SATURATION LOGIC LIMITS THE INTEGRATOR CONTENT TO NUMBERS         DIG01780
C      REPRESENTING WHEEL SPEEDS OF 900 TO 1300 RPM APPROXIMATELY       DIG01790
C THE WHEEL SPEED REGISTER CONTENT REPRESENTS ONE FOURTH OF A DESIRED   DIG01800
C      REVOLUTION. PERIOD RESOLVED TO 0.1 MSEC                          DIG01810
C UPPER THRESHOLD 1300 RPM                                              DIG01820
      IRNP=((ISP)*IDINT(XP1)+IDINT((DFLOAT(IQNP))*IXP2)))               DIG01830
      IF( MOD(N,NP).EQ.0)IQNP=IQNP+ISP                                  DIG01840
      IF(IQNP.GT.QNPMAX)IQNP=QNPMAX                                     DIG01850
      IF(IQNP.LT.QNPMIN)IQNP=QNPMIN                                     DIG01860
      IWM=IQNP-IINT                                                     DIG01870
C     WRITE(6,617)T,ISP,IWM                                             DIG01880
  617 FORMAT(1X,' TIME=',1PE13.6,' SECONDS ','ISP=',I6,'  IQNP=',I6,//) DIG01890
      GO TO 68                                                          DIG01900
C MODEP=4 IS THE PITCH BACK UP MODE
78    IF(MOD(N,INP).NE.0) GO TO 68
      ISP=0
      DO 79 I=1,16
79    ISP=ISP+ISAMP(I)
      ISP=ISP/16
      IDAPB=(ISP-ISPO)/(INP*4)
      ISPO=ISP
      ISIGB=IDAPB*IDINT(XP4)+ISP
      IF(ISIGB.GT.IDELB)ISEQU(1)=5
      IF(ISIGB.LT.-1*IDELB)ISEQU(1)=1
      IF(IABS(ISIGB).LE.IDELB)ISEQU(1)=0
38    IRNP=IINI*XP2
68    IS=0                                                              DIG01920
      RETURN                                                            DIG01930
      END                                                               DIG01940
```

****

```
      SUBROUTINE OUTPUT (T,Y,TERM)                                      TEM03990
      IMPLICIT REAL*8(A-H,O-Z)                                          TEM04000
      DIMENSION Y(1)                                                    TEM04010
      REAL*4 PX(5001),PY1(5001),PY2(5001),PY3(5001),PY4(5001),PY5(5001) TEM04020
      REAL*4 SP(5001,2)
      REAL*4 PY7(5001)
      COMMON /TEMP/ PY7
      COMMON /BETTER/ S(2),SP
      LOGICAL TERM                                                      TEM04030
     0COMMON /GR/DUM(3),NPL                                             TEM04040
     1       /OTPT/PX,PY1,PY2,PY3,PY4,PY5                               TEM04050
      DATA RAD /.0174532925/
C                                                                       TEM04060
C                                                                       TEM04070
      NPL=NPL+1                                                         TEM04080
C
C     PICK UP THE INITIAL WHEEL SPEED BIAS
C
      IF(NPL.EQ.1)WHSD0=Y(5)/.0068055
      IF (NPL.LT.5001) GO TO 20                                         TEM04090
      TERM=.TRUE.                                                       TEM04100
      WRITE (6,10)                                                      TEM04110
   10 FORMAT ('0 PLOT ARRAYS FILLED COMPLETELY.')                       TEM04120
   20 PX(NPL)=T                                                         TEM04130
      PY1(NPL)=Y(1)/RAD                                                 TEM04140
      PY2(NPL)=Y(2)/RAD                                                 TEM04150
      PY3(NPL)=Y(3)/RAD                                                 TEM04160
      PY4(NPL)=Y(4) /RAD                                                TEM04170
      SP(NPL,1)=S(1)/RAD
      SP(NPL,2)=S(2)/RAD
C
C     SUBTRACT THE INITIAL WHEEL SPEED BIAS WHSD0
C
      PY5(NPL)=Y(5)/.0068055-WHSD0
      PY7(NPL)=Y(6)/RAD
      RETURN                                                            TEM04190
      END                                                               TEM04200
```

****

```
      SUBROUTINE AM(H,T,Y,FY,GLB,LUB,TERM)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION COR(20),YDOT(20),Y(20),YP(20),FY(20,5),P(4),C(4)
      INTEGER HLV,DBL
      LOGICAL TERM
      REAL*8 LUB,MAX, K
      COMMON /COEF/P,C,TBOUND,WR,MPT,NCT,NSTEP, NEQ
      COMMON /ANGMOM/HSQ0
      COMMON /PARAM/XJ(4),XJ23,XJ31,XJ21,T1,T2,OMEGA0,XP1,XP2,XD1,XD2,
     2              XD3,XR1,XR2
      COMMON /ELLEN/ TIMDEL,TIM,SIGMA,IB,NP,MODER,NR,IRNP,IRNR,MODEP,
     2IDEL,MMODEP,IPNP,XP3,LSB
      DATA DBL,HLV/2*0/
C
C
      ERSCAL = 19.0D+00/270.0D+00
10    CALL TORQUE(Y)
      CALL DERIV(Y,YDOT)
      DO 20 I=1, NEQ
   20 FY(I,4)=YDOT(I)
C
C CALCULATE PREDICTOR
      DO 30 I=1, NEQ
   30 YP(I)=Y(I)+H*(P(1)*FY(I,1)+P(2)*FY(I,2)+P(3)*FY(I,3)+P(4)*FY(I,4))
C
C CALCULATE CORRECTOR
      CALL DERIV(YP,YDOT)
      DO 40 I=1, NEQ
   40 FY(I,5)=YDOT(I)
      DO 50 I=1, NEQ
   50 Y(I)=Y(I)+H*(C(1)*FY(I,2)+C(2)*FY(I,3)+C(3)*FY(I,4)+C(4)*FY(I,5))
      T=T+H
      TIM=T
      NSTEP=NSTEP+1
      NCT=NCT+MPT
C
C SEE IF IT'S TIME TO PLOT, WRITE, OR TERMINATE.
      IF (NCT.LT.256) GO TO 53
      NCT=0
      CALL OUTPUT(T,Y,TERM)
      IF (TERM) GO TO 130
   53 NWR=T/WR+1
      IF (DABS(T-NWR*WR).GT.H*2.D0) GO TO 57
      WRITE(6,55)T,(Y(I),I=1,11),NSTEP,HLV,DBL
   55 FORMAT (/' AT T=',F7.2/' Y=',11(G10.3,',' )/,' NSTEP=',I6,' HLV=',
     1 I5,', DBL=',I5/)
      X=Y(5)*(Y(6)*Y(4)+Y(8))
      THEDOT=(-1./XD1)*(XD2*Y(4)+XD3*Y(11)-XR1*Y(2)-XR2*Y(10)+X)
      DHSQ=((XJ(1)*Y(6)+Y(5)*DCOS(Y(4)))**2
     1     + XJ(2)*Y(7)**2
     2     +(XJ(3)*Y(8) -Y(5)*DSIN(Y(4)))**2
     3     -HSQ0)/HSQ0
      WRITE (6,56) DHSQ
   56 FORMAT (' CHANGE IN SQUARE OF ANGULAR MOMENTUM = ',G10.3)
   57 IF (T.GE.TBOUND) GO TO 130
C
C SEE IF STEP SIZE SHOULD BE CHANGED.
      DO 70 I=1, NEQ
      IF (DABS(Y(I)).LE.5.D-6) GO TO 60
```

```
      COR(I)=DABS((Y(I)-YP(I))/Y(I))                                   TEM01790
      GO TO 70                                                          TEM01800
   60 COR(I)=0.D0                                                       TEM01810
   70 CONTINUE                                                          TEM01820
      MAX=COR(1)                                                        TEM01830
      DO 90 I=2, NEQ                                                    TEM01840
      IF (MAX-COR(I)) 80,90,90                                          TEM01850
   80 MAX=COR(I)                                                        TEM01860
   90 CONTINUE                                                          TEM01870
      MAX = MAX*ERSCAL                                                  TEM01880
      IF (MAX.LT.GLB) GO TO 100                                         TEM01890
      IF (MAX.LE.LUB) GO TO 110                                         TEM01900
C                                                                       TEM01910
C HALVE H.                                                              TEM01920
      IF (MPT.EQ.1) GO TO 110                                           TEM01930
      H=H/2.D0                                                          TEM01940
      MPT=MPT/2                                                         TEM01950
      HLV=HLV+1                                                         TEM01960
      CALL RK(H,T,Y,FY,TERM)                                            TEM01970
      IF (TERM) GO TO 130                                               TEM01980
      GO TO 10                                                          TEM01990
C                                                                       TEM02000
C DOUBLE H.                                                             TEM02010
C THE 32 IN FOLLOWING CARD WAS PUT IN FOR THE DIGITAL SYSTEM TO LIMIT
C    THE NUMBER OF TIMES THE STEP SIZE COULD BE DOULBLED (WAS 256)
  100 IF (MPT.EQ.32 .OR. MOD(NCT/MPT,2).NE.0) GO TO 110                 TEM02020
      H=H*2.D0                                                          TEM02030
      MPT=MPT*2                                                         TEM02040
      DBL=DBL+1                                                         TEM02050
      CALL RK(H,T,Y,FY,TERM)                                            TEM02060
      IF (TERM) GO TO 130                                               TEM02070
      GO TO 10                                                          TEM02080
C                                                                       TEM02090
C H UNCHANGED. SHIFT TO FIND NEW SET OF DERIVATIVES.                    TEM02100
  110 DO 120 J=1,3                                                      TEM02110
      DO 120 I=1, NEQ                                                   TEM02120
  120 FY(I,J)=FY(I,J+1)                                                 TEM02130
      GO TO 10                                                          TEM02140
C                                                                       TEM02150
C TERMINATE INTEGRATION. WRITE AND PUNCH FINAL VALUES.                  TEM02160
130   WRITE(6,55)T,(Y(I),I=1,11),NSTEP,HLV,DBL
C     WRITE (7,140) T,Y                                                 TEM02180
  140 FORMAT (Z16)                                                      TEM02190
      NCT=0                                                             TEM02200
      MPT=16                                                            TEM02210
      DBL=0                                                             TEM02220
      HLV=0                                                             TEM02230
      RETURN                                                            TEM02240
      END                                                               TEM02250
```

****

```
      SUBROUTINE RK(H,T,Y,FY,TERM)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(1),FY(20,5),YDOT(20),G1(20),G2(20),G3(20),G4(20),Z1(20
     1), Z2(20),Z3(20),PHI(20)
      LOGICAL TERM
      COMMON /COEF/DUM(8),TBOUND,WR,MPT,NCT,NSTEP, NEQ
      COMMON/ELLEN/ TIMDEL,TIM,SIGMA,IB,NP,MODER,NR,IRNP,IRNR,MODEP,
     2IDEL,MMODEP,IPNP,XP3,LSB
C
C
      H2=H/2.D0
      DO 50 M=2,4
      CALL TORQUE (Y)
      CALL DERIV(Y,YDOT)
      DO 10 I=1, NEQ
      FY(I,M-I)=YDOT(I)
      G1(I)=YDOT(I)
   10 Z1(I)=Y(I)+G1(I)*H2
C
      CALL DERIV(Z1,YDOT)
      DO 20 I=1, NEQ
      G2(I)=YDOT(I)
   20 Z2(I)=Y(I)+G2(I)*H2
C
      CALL DERIV(Z2,YDOT)
      DO 30 I=1,NEQ
      G3(I)=YDOT(I)
   30 Z3(I)=Y(I)+G3(I)*H
C
      CALL DERIV(Z3,YDOT)
      DO 40 I=1, NEQ
      G4(I)=YDOT(I)
      PHI(I)=(G1(I)+2.D0*G2(I)+2.D0*G3(I)+G4(I))/6.D0
   40 Y(I)=Y(I)+H*PHI(I)
      T=T+H
      TIM=T
      NSTEP=NSTEP+1
      NCT=NCT+MPT
      IF (NCT.LT.256) GO TO 50
      NCT=0
      CALL OUTPUT(T,Y,TERM)
      IF (TERM) RETURN
   50 CONTINUE
      RETURN
      END
```

****

```
      SUBROUTINE CR2(X,Y)                                               TEM03740
      DIMENSION X(1),Y(1)                                               TEM03750
      COMMON /SKIP/ IERR
      INTEGER VMPH,HMPH,VLBL,HLBL,VCHR,HCHR,BEGIN                       TEM03760
      COMMON /GR/YL,XR,YB,YT,VLN,HLN,NPL,N,NPT,BEGIN,INC                TEM03770
      DATA DC/20./
C                                                                       TEM03790
      CALL DXDYV(1,XL,XR,DX,NX,I,NNX,DC,IERR)
      IF(IERR.NE.0) RETURN
      CALL DXDYV(2,YB,YT,DY,M,J,NY,DC,IERR)
      IF(IERR.NE.0) RETURN
      CALL GRID1V(1,XL,XR,YB,YT ,DX,DY,NX,M,I,J,NNX,NY)
      CALL APLOTV(NPT*INC,X(BEGIN),Y(BEGIN),INC,INC,1,44,IER)           TEM03810
      DO 1 I=BEGIN,N,INC                                                TEM03820
      CALL LINEV(NXV(X(I)),NYV(Y(I)),NXV(X(I+INC)),NYV(Y(I+INC)))       TEM03830
    1 CONTINUE                                                          TEM03840
      RETURN                                                            TEM03850
      END                                                               TEM03860
```

****

```
      SUBROUTINE UNLOAD (Y)                                             UNL00010
C UNLOADING SIMULATION LES 8/9                                          UNL00020
C                                                                       UNL00030
C COMPUTES THE FUNCTION ASEQU (ISEQU)                                   UNL00040
C  ASEQU IS THE CONTROL LINK TO SUBROUTINES TORK AND TORQUE             UNL00050
C                                                                       UNL00060
      IMPLICIT REAL* 8 (A-H,O-Z)                                        UNL00070
      REAL *8 Y(1)                                                      UNL00080
      COMMON /SEQU/ DUM(3),ISEQU(3)                                     UNL00090
      COMMON /DUMP/ Y4MAX,Y5MIN,Y5MAX,Y2MAX,YO50,MODE
      COMMON/ELLEM/DUM1(4),MODER,IDUM(3),MODEP                          UNL00120
C                                                                       UNL00130
      GO TO (10,20,10),MODER                                            UNL00140
10    GO TO (40,30),MODE                                                UNL00150
C MODE=1 NORMAL OPERATION                                               UNL00160
C MODE=2 ROLL BACK-UP CONTROL                                           UNL00170
40    GO TO (50,50,60,95),MODEP
60    ISEQU(1)=0                                                        UNL00190
90    ISEQU(2)=0                                                        UNL00200
95    RETURN
C                                                                       UNL00220
C ***************                                                       UNL00230
C                                                                       UNL00240
 20   I3=ISEQU(3)                                                       UNL00250
      IF(I3)21,21,24                                                    UNL00260
21    IF(DABS(Y(4)).LT.Y4MAX) GO TO 40                                  UNL00270
      IF(Y(4).GT.Y4MAX) GO TO 23                                        UNL00280
      ISEQU(3)=7                                                        UNL00290
      GO TO 40                                                          UNL00300
23    ISEQU(3)=3                                                        UNL00310
      GO TO 40                                                          UNL00320
24    IF(I3-3) 25,25,26                                                 UNL00330
25    IF(Y(4).LE.0) ISEQU(3)=0                                          UNL00340
      GO TO 40                                                          UNL00350
26    IF(Y(4).GE.0) ISEQU(3)=0                                          UNL00360
      GO TO 40                                                          UNL00370
C                                                                       UNL00380
C ***********                                                           UNL00390
C                                                                       UNL00400
 30   IF(ISEQU(3))31,31,34                                              UNL00410
31    IF(DABS(Y(2)).LT.Y2MAX) GO TO 40                                  UNL00420
      IF(Y(2).GE.Y2MAX) GO TO 33                                        UNL00430
      ISEQU(3)=7                                                        UNL00440
      GO TO 40                                                          UNL00450
33    ISEQU(3)=3                                                        UNL00460
      GO TO 40                                                          UNL00470
34    IF(ISEQU(3)-3) 35,35,36                                           UNL00480
35    IF(Y(2).LE.0) ISEQU(3)=0                                          UNL00490
      GO TO 40                                                          UNL00500
36    IF(Y(2).GE.0) ISEQU(3)=0                                          UNL00510
      GO TO 40                                                          UNL00520
C                                                                       UNL00530
C *********                                                             UNL00540
C                                                                       UNL00550
 50   IF(ISEQU(1))51,51,52                                              UNL00560
51    IF(DABS(Y(5)-YO50).LT.DABS(YO50-Y5MIN))GO TO 90                   UNL00570
      IF(Y(5).GT.Y5MAX) ISEQU(1)=5
      IF(Y(5).LT. Y5MIN) ISEQU(1)=1
      IF(ISEQU(1).EQ.0) WRITE(6,100)Y(5),Y5MIN,Y5MAX,YO50               UNL00600
```

```
100   FORMAT(' ERRORO  THERE IS NO TRUE COMBINATION FOR Y(5)'/' Y(5)=',EUNL00610
     120.8,'Y5MIN=',E20.8,' Y5MAX=',E20.8,' Y050=',E20.8)              UNL00620
      GO TO 90                                                          UNL00630
52    IF(ISEQU(1)-1) 53,53,54                                           UNL00640
53    IF(Y(5)-Y050.LE.0) ISEQU(1)=0                                     UNL00650
      GO TO 90                                                          UNL00660
54    IF(Y(5)-Y050.GE.0) ISEQU(1)=0                                     UNL00670
      GO TO 90                                                          UNL00680
      END                                                               UNL00690
```

****

```
C
C
      SUBROUTINE TURK(AA,BB)
C
C    AA IS THE THRUSTER LOCATION COORDINATE ON Z1, BB IS THE LOCATION
C      COORDINATE ON Z2.  THIS SUBROUTINE ESTABLISHES A TORQUE
C     LOOK-UP TABLE FIRST BY FILLING IN THE SIGNS (+ OR -) OF THE
C     TORQUE COMPONENTS FOR EACH THRUSTER IN A 12 X 3 ARRAY.  IT
C      THEN COMPUTES THE MAGNITUDES OF THE TORQUES AND REPLACES EACH
C     ENTRY  WITH THE TORQUE COMPONENTS ALONG THE Z1, Z2, AND Z3 AXES.
C
      IMPLICIT REAL*8(A-H,O-Z)
         INTEGER*4  TRUST1,TRUST2,TRUST3,TRUST4,J
      REAL*8 TABLE(12,3),L1,L2(2), L3(2), AA(2), T(3), AF(2)
      COMMON /CONTRL/ T
      DATA TABLE/-1.,1.,-1.,1.,1.,-1.,1.,-1.,0.,0.,0.,0.,
     C           1.,-1.,-1.,1.,-1.,1.,1.,-1.,0.,0.,0.,0.,
     C           -1.,-1.,1.,1.,-1.,-1.,1.,1.,1.,-1.,-1.,1. /,
     C           THETA,PHI,PI/30.,5.,3.1415926535897/,FORCE/.000036/
C
C     CONVERT FROM DEGREES TO RADIANS AND FROM INCHES TO FEET
C
         THETR = THETA * (PI/180.)
         PHR = PHI * (PI/180.)
      AF(1) = AA(1)/12.0D+00
      AF(2) = AA(2)/12.0D+00
      BF = BB/12.0D+00
         L1 = BF * FORCE * DSIN(THETR)
      DO 8  I = 1 , 2
      L2(I) = AF(I)*FORCE*DSIN(THETR)
    8 L3(I) = AF(I)*FORCE*DCOS(THETR)
         BFSIN = BF * FORCE * DSIN(PHR)
      J = 1
         DO 9 I=1,8
            TABLE(I,1) = DSIGN(L1,TABLE(I,I))
      TABLE(I, 2) = DSIGN(L2(J), TABLE(I, 2))
      TABLE(I, 3) = DSIGN(L3(J), TABLE(I, 3))
      J = J + 1
      IF( J .GE. 3 )  J = 1
    9    CONTINUE
            TABLE(9,3) = BFSIN
            TABLE (10,3) = -BFSIN
            TABLE(11,3) = -BFSIN
            TABLE(12,3) = BFSIN
       WRITE(6,100)((TABLE(I,J),J=1,3),I=1,12)
  100  FORMAT('1TORQUE TABLE IS SET UP AS FOLLOWS0'//,
     C 12(10X,D12.5,10X,D12.5,10X,D12.5/))
         RETURN
C
C
      ENTRY THRUST(TRUST1,TRUST2,TRUST3,TRUST4)
C
C   THIS IS THE ACTIVE PART OF THE SUBROUTINE, TRUST1-4 ARE
C   NON-NEGATIVE INTEGERS WHICH SIGNIFY WHICH AND HOW MANY THRUSTERS ARE
C   TO BE TURNED ON.  TRUST1 IS ASSUMED TO BE ZERO IF NO THRUSTERS ARE
C   TO BE FIRED.  THE SUBROUTINE ADDS THE COMPONENT TORQUES OF THE
C   THRUSTERS FIRED TO THE TOTAL TORQUE COMPONENTS  STORED IN COMMON.
C
         IF(TRUST1 .LE. 0) GO TO 16
```

```
      ASSIGN 11 TO J                                            TEM05220
      K = TRUST1                                                TEM05230
      GO TO 14                                                  TEM05240
11    IF(TRUST2 .LE. 0) GO TO 18                                TEM05250
      ASSIGN 12 TO J                                            TEM05260
      K = TRUST2                                                TEM05270
      GO TO 14                                                  TEM05280
12    IF(TRUST3 .LE. 0) GO TO 18                                TEM05290
      ASSIGN 13 TO J                                            TEM05300
      K = TRUST3                                                TEM05310
      GO TO 14                                                  TEM05320
13    IF(TRUST4 .LE. 0) GO TO 18                                TEM05330
      ASSIGN 18 TO J                                            TEM05340
      K = TRUST4                                                TEM05350
14    DO 15 I=1,3                                               TEM05360
      T(I) = TABLE(K,I) + T(I)                                  TEM05370
15    CONTINUE                                                  TEM05380
      GO TO J,(11,12,13,18)                                     TEM05390
16    WRITE(6,17)                                               TEM05400
17    FORMAT(' ENTRY POINT THRUST CALLED UNNECESSARILY.')       TEM05410
18 CONTINUE                                                     TEM05420
   RETURN                                                       TEM05430
      END                                                       TEM05440
                    ************************
```

```
      SUBROUTINE TORQUE (Y)
      REAL*8 Y(1)
      REAL*8 TTIME, H, TLIMIT, ATIME, ALIMIT, L(3)                      TEM05460
      INTEGER*4 ASEQU(3), AXIS,    TCUR, APOINT, SELECT(8, 4)           TEM05470
      COMMON /CONTRL/ L                                                 TEM05480
     2       /SEQU/ H, ATIME, TTIME, ASEQU, APOINT                      TEM05490
     3       /TABLES/ SELECT                                            TEM05500
      DATA TLIMIT, ALIMIT /1.0D+00, 1.0D+02/                            TEM05510
C                                                                       TEM05520
C     SET TORQUES TO ZERO                                               TEM05530
C                                                                       TEM05540
      CALL UNLOAD (Y)
      DO 100  I = 1, 3                                                  TEM05550
  100 L(I) = 0.0D+00                                                    TEM05560
      IF( APOINT .GE. 4 )  RETURN                                       TEM05570
C                                                                       TEM05580
C     HAVE WE BEEN FIRING ABOUT THIS AXIS MORE THAN ALIMIT              TEM05590
C                                                                       TEM05600
      IF( ATIME .LT. ALIMIT )  GO TO 200                                TEM05610
C                                                                       TEM05620
C     SELECT NEW AXIS                                                   TEM05630
C                                                                       TEM05640
      APOINT = APOINT + 1                                               TEM05650
      IF(APOINT.GT.3)APOINT=1
      AXIS = ASEQU(APOINT)                                              TEM05670
      IF( ( AXIS .LT. 1 ) .OR. ( AXIS .GT. 8 ) )  RETURN                TEM05680
      ATIME = 0.0D+00                                                   TEM05690
      TCUR = 1                                                          TEM05700
      GO TO 210                                                         TEM05710
C                                                                       TEM05720
C     HAVE WE BEEN FIRING THIS THRUSTER FOR MORE THAN TLIMIT            TEM05730
C                                                                       TEM05740
  200 IF( TTIME .LT. TLIMIT )  GO TO 300                                TEM05750
C                                                                       TEM05760
C     SELECT NEW THRUSTER                                               TEM05770
C                                                                       TEM05780
      TCUR = TCUR + 1                                                   TEM05790
      IF( (TCUR .GT. 4) .OR. (SELECT(AXIS,  TCUR) .LT. 1) )  TCUR = 1   TEM05800
  210 IFIRE = SELECT(AXIS,   TCUR)                                      TEM05810
      TTIME = 0.0D+00                                                   TEM05820
C                                                                       TEM05830
C     SET UP CALL TO THRUST AND INCREMENT TIMES                         TEM05840
C                                                                       TEM05850
  300 CALL THRUST(IFIRE, 0, 0, 0)                                       TEM05860
      ATIME = ATIME + H                                                 TEM05870
      TTIME = TTIME + H                                                 TEM05880
  301 RETURN
      END                                                               TEM05900
                         *************************
```

```
      BLOCK DATA                                                          TEMO2260
      IMPLICIT REAL*8 (A-H,O-Z)                                           TEMO2270
      REAL*8 K                                                            TEMO2280
      REAL*8 M2,M3
      COMMON /ALIGN/ M2,M3
      COMMON /ELLEN/ TIMDEL,TIM,SIGMA,IB,NP,MODER,NR,IRNP,IRNR,MODEP,
     1IDEL,MMODEP,IPNP,XP3,LSB
      DIMENSION P(4),C(4),XJ(4)                                           TEMO2290
     0COMMON /COEF/P,C,TBOUND,WR,MPT,NCT,NSTEP, NEQ                       TEMO2300
     1        /GR/DUM(3),NPL,IDUM(3),INC                                  TEMO2310
     2           /PARAM/XJ,DUM1(3),T1,T2,OMEGA0,XP1,XP2,XD1,XD2,XD3,XR1,  TEMO2320
     3           XP2                                                      TEMO2330
     3           /SEQU/ DUM2(3), ISEQU(3), IPOINT                         TEMO2340
     4           /TABLES/ ISEL(8, 4)                                      TEMO2350
      COMMON/PITCH/XP4,IDELB,INP
C                                                                         TEMO2360
C     INTIALIZE      /PARAM/                                              TEMO2370
C                                                                         TEMO2380
      DATA XJ /120.,130.,20.,.065/
      DATA XP1,XP2,XD1,XD2,XD3,XR1,XR2/1.,.125,6.,.3,0.,.125,1.0/
C                                                                         TEMO2420
C     INITIALIZE      /COEF/                                              TEMO2430
C                                                                         TEMO2440
      DATA MPT,NCT,NSTEP,NEQ/16,2*0,12/
C                                                                         TEMO2460
C     INITIALIZE      /GR/                                                TEMO2470
C                                                                         TEMO2480
      DATA NPL, INC /0, 1/                                                TEMO2490
C                                                                         TEMO2500
C     INITIALIZE      /SEQU/                                              TEMO2510
C                                                                         TEMO2520
      DATA IPOINT,ISEQU/0,0,0,0/,DUM2(3), DUM2(3)/2*1.0D+03/
      DATA DUM2(2) /1.0D+03/
C                                                                         TEMO2540
      DATA TBOUND/600./
      DATA SIGMA /0.0/
      DATA NR /2/
      DATA MODER/2/
      DATA MODEP /2/
      DATA MMODEP /200000/
      DATA IDEL /24/
      DATA XP4,IDELB,INP/200.,20,5/
      DATA TIMDEL /0.0/
      DATA NP /3/
      DATA M2/0.0/
      DATA M3/0.0/
      DATA XP3/16./
      DATA LSB /500000/
      DATA XP3/6./
C     INITIALIZE      /SELECT/                                            TEMO2550
C                                                                         TEMO2560
      DATA ISEL /2, 1, 3, 11, 1, 2, 2,  9,                                TEMO2570
     2           7, 7, 7, 12, 8, 8, 6, 10,                                TEMO2580
     3           0, 0, 0,  0, 0, 0, 0,  0,                                TEMO2590
     4           0, 0, 0,  0, 0, 0, 0,  0/                                TEMO2600
      END                                                                 TEMO2610
```

****

```
      SUBROUTINE GR1(FR,X,Y)                                            TEM03100
      DIMENSION X(1),Y(1)                                               TEM03110
      INTEGER BEGIN,FR                                                  TEM03120
      COMMON /GR/XL,XR,YB,YT,DX, DY, NPL,NDO,NPT,BEGIN,INC
C                                                                       TEM03140
C FIND UPPER AND LOWER GRID LIMITS                                      TEM03150
      CALL MINMAX(Y,NPL,YB,YT)                                          TEM03160
      IF (YB) 10,30,20                                                  TEM03170
   10 YB=1.05*YB                                                        TEM03180
      GO TO 30                                                          TEM03190
   20 YB=0.95*YB                                                        TEM03200
   30 CONTINUE                                                          TEM03210
      IF (YT) 40,60,50                                                  TEM03220
   40 YT=0.95*YT                                                        TEM03230
      GO TO 60                                                          TEM03240
   50 YT=1.05*YT                                                        TEM03250
   60 CONTINUE                                                          TEM03260
C                                                                       TEM03270
      BEGIN=1                                                           TEM03280
      XL=X(1)                                                           TEM03290
      DY=(YT-YB)/30.                                                    TEM03300
C                                                                       TEM03310
C IS MORE THAN ONE FRAME DESIRED                                        TEM03320
      IF (FR.GT.1) GO TO 80                                             TEM03330
C ENTIRE GRAPH TO BE PLOTTED ON ONE FRAME.                              TEM03340
      NPT=NPL                                                           TEM03350
      XR=X(NPL)                                                         TEM03360
      DX=(XR-XL)/20.                                                    TEM03370
      NDO=BEGIN+(NPT-1)*INC-1                                           TEM03410
      CALL GR2(X,Y)                                                     TEM03420
      RETURN                                                            TEM03430
C                                                                       TEM03440
C GRAPH TO BE PLOTTED ON MORE THAN ONE FRAME.                           TEM03450
   80 IFR1=FR-1                                                         TEM03460
      NPLFR=(NPL+IFR1)/FR                                               TEM03470
      IF (MOD(NPL+IFR1,FR).NE.0) NPLFR=NPLFR+1                          TEM03480
      NPT=NPLFR                                                         TEM03490
      NM1=NPLFR-1                                                       TEM03500
      DX=(X(NPT)-XL)/20.                                                TEM03510
C                                                                       TEM03540
C PLOT INITIAL FRAMES.                                                  TEM03550
      DXR=20.*DX                                                        TEM03560
      XR=XL+DXR                                                         TEM03570
      DO 90 I=1,IFR1                                                    TEM03580
      IX=I                                                              TEM03590
      NDO=BEGIN+(NPT-1)*INC-1                                           TEM03600
      CALL GR2(X,Y)                                                     TEM03610
      XL=XR                                                             TEM03620
      XR=XR+DXR                                                         TEM03630
      BEGIN=BEGIN+NM1                                                   TEM03640
      IF (XR.GE.X(NPL)) GO TO 100                                       TEM03650
   90 CONTINUE                                                          TEM03660
C                                                                       TEM03670
C PLOT FINAL FRAME.                                                     TEM03680
  100 NPT=NPL-IX*NM1                                                    TEM03690
      NDO=BEGIN+(NPT-1)*INC-1                                           TEM03700
      CALL GR2(X,Y)                                                     TEM03710
      RETURN                                                            TEM03720
      END                                                               TEM03730
```

```
      SUBROUTINE MINMAX (X, N, XMIN, XMAX )
      DIMENSION X(1)
      XMIN = X(1)
      XMAX = X(1)
      DO 10 I=2,N
      IF (X(I)-XMIN) 1,2,2
    1 XMIN = X(I)
    2 IF (XMAX-X(I)) 3,10,10
    3 XMAX = X(I)
   10 CONTINUE
      RETURN
      END
```

****

DOCUMENT CONTROL DATA – R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Lincoln Laboratory, M.I.T.	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP None

3. REPORT TITLE

LES-8/9 Attitude Control System Numerical Simulation

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Technical Note

5. AUTHOR(S) *(Last name, first name, initial)*

Floyd, Franklin W.   Much, Carl H.   Smith, Norval P.   Swenson, Ellen H.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
5 February 1971	58	5

8a. CONTRACT OR GRANT NO. F19628-70-C-0230	9a. ORIGINATOR'S REPORT NUMBER(S) Technical Note 1971-10
b. PROJECT NO. 649L c. d.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* ESD-TR-71-11

10. AVAILABILITY/LIMITATION NOTICES

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
None	Air Force Systems Command, USAF

13. ABSTRACT

A program was written to simulate the three axis attitude control system of LES-8/9. The underlying theory to the computer simulation and detailed outlines of each of the subroutines involved in the complete program are described.

Whenever feasible, the simulation was written to duplicate as closely as possible the logic and signal flow of the actual digital attitude control system. Each subroutine was thoroughly verified as accurate by independent and integral system operation, as well as by theoretical estimates, analog computer simulations and actual experimental data.

A substantial compilation of data from this working program was catalogued and analyzed for attitude control system evaluation and optimization. This program also proved itself to be invaluable in the analysis of stability and performance of the complete attitude control system.

14. KEY WORDS

LES	digital attitude control system
analog computer simulation	satellite